A Mini Project Report On

Wireless Network Intrusion Detection System

Submitted in Partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

In

Department of Computer Science and Engineering

By

G. Sai Aditya	–	**22241A0532**
K. Naveen	–	**22241A0540**
B. Lok Vaedhan	–	**22241A0514**
K. Sai Charan	–	**22241A0537**
B. Vamshi	–	**22241A0513**

Under the Esteemed guidance of

Dr. P. Vara Prasada Rao

Associate Professor

Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Bachupalli, Kukatpally ,Hyderabad,Telangana,500090

2024-2025

GOKARAJU RANGARAJU

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

CERTIFICATE

This is to certify that the Mini Project entitled "**Wireless Network Intrusion Detection System**" is submitted by **G. Sai Aditya (22241A0532), K. Naveen (22241A0540), B. lok Vardhan (22241A0514), K. Sai Charan (22241A0537), B. Vamshi (22241A0513)** Partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in Computer Science and Engineering during the academic year **2024-2025.**

INTERNAL GUIDE

Dr. P. Vara Prasada Rao

Professor

HEAD OF THE DEPARTMENT

Dr. B. SANKARA BABU

Professor & HoD

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Many people helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First, we wish to express our deep gratitude to our internal guide **Dr. P. Vara Prasada Rao,Associate Professor**, Department of CSE for his support in the completion of our project report. We wish to express our honest and sincere thanks to **V. Jyothi** and **K. Adilakshmi** for coordinating in conducting the project reviews. We express our gratitude to **Dr. B. Sankara Babu, HOD**, department of CSE for providing resources, and to the principal **Dr. J. Praveen** for providing the facilities to complete our Mini Project. We would like to thank all our faculty and friends for their help and constructive criticism during the project completion phase. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

G. Sai Aditya (22241A0532)
K. Naveen (22241A0540)
B. Lok Vardhan (22241A0514)
K. Sai Charan (22241A0537)
B. Vamshi (22241A0513)

DECLARATION

We hereby declare that the Mini Project entitled "**Wireless Network Intrusion Detection System** " is the work done during the period from 16th Jan 2025 to 13th May 2025 and is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering from **Gokaraju Rangaraju Institute of Engineering and Technology**. The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

G.Sai Aditya (22241A0532)

K. Naveen (22241A0540)

B. Lok Vardhan (22241A0514)

K. Sai Charan (22241A0537)

B. Vamshi (22241A0513)

ABSTRACT:

- Wireless networks are highly vulnerable to cyber attacks that can breach the network integrity, intercept confidential data, and break connectivity. Common attacks like Deauthentication Attacks, Evil Twin Attacks, ARP Spoofing, and Wi-Fi-based DDoS Attacks threaten individual and business networks highly. These attacks can cause unauthorized access, data breach, and service denial, and hence call for proactive security measures.
- In order to overcome such limitations, this project suggests implementing a Wireless Intrusion Detection and Response System (WIDRS) on Raspberry Pi or Ubuntu operating systems. The system constantly monitors and examines wireless network traffic with high-capacity security tools like Aircrack-ng, iptables, Fail2Ban, Snort, and tcpdump. When the system detects suspicious behavior, it engages automatic response actions like sending notifications through email and Telegram, blocking attack IPs, and deauthenticating malicious users to prevent them from being further exploited
- This solution performs exceptionally well in public Wi-Fi networks, business networks, home networks, and IoT networks with real-time monitoring, superior network visibility, and enhanced cybersecurity. With automated threat detection and response, WIDRS strengthens network protection, offering a more secure and robust wireless communication environment

CHAPTER 1

INTRODUCTION

In a time of fast-paced technological advancements and ubiquitous connectivity, wireless networks have emerged as a cornerstone of modern communications networks. Wireless networks make easy information sharing possible, support real-time collaboration, and provide mobility in ways unseen before. From homes and homes for learning to businesses and government, wireless access is no longer a luxury but a necessity. But growing wireless technology usage has also opened doors to a vast array of security vulnerabilities and loopholes. Unlike wire-based networks where access is physically limited, wireless networks employ open airwaves and thus are inherently more vulnerable to cyberattacks.

The broadcast nature of wireless communication renders the information susceptible to being easily eavesdropped on, compromised, or injected with unauthorized data packets without any physical hardware access within the network needed. Such ease of exploitation has given birth to a sharp increase in wireless-specific threats like eavesdropping, rogue access points, session hijacking, denial-of-service (DoS) attacks, and unauthorized outsider access. Furthermore, as wireless networks become increasingly sophisticated—frequently integrating Internet of Things (IoT) devices, mobile devices, and cloud-connected services—the attack surface increases exponentially. Standard security controls like firewalls and antivirus products, though still needed, are usually not enough to detect or respond to sophisticated and evasive attacks in the wireless realm. This immediately calls for advanced systems that not only alert on intrusions but also respond to them in real time.

To fulfill this need, the idea of a Wireless Intrusion Detection and Response System (WNIDRS) has emerged as a credible line of defense. A WNIDRS is an enterprise security appliance that continuously scans the wireless communications network, identifies unauthorized or suspicious traffic, and responds against attacks before they can destroy the confidentiality, integrity, or availability of the network. In contrast to passive threat surveillance systems, WNIDRS is both a detector and a responder and is therefore best suited

to rapidly evolving threat-impacted environments. It is best used in wireless environments where threats are more difficult to detect and tend to rapidly propagate on attached devices.

At the base of WNIDRS is a dual-detection mechanism: signature-based detection and anomaly-based detection. Signature-based detection verifies real-time traffic against a repository of known attacks or signatures. As soon as a match is found, the system sends an immediate alarm, enabling timely detection of known attacks. The method is good and speedy in detecting well-documented threats but is bad where new or emerging threats are involved. In order to address this deficiency, anomaly-based detection is utilized. In this, the norm of "normal" network behavior is derived by the usage of statistical methods and machine learning methods. Anything that is out of place with respect to the established norm—i.e., abnormal bursts in traffic, puzzling device behavior, or non-adjacent access—is selected to be studied. If both approaches are used together, WNIDRS correctly identifies with the least amount of false positives.

In actual field deployments, WNIDDRS employs hardware sensors and software agents to execute its functions. Two of the most widely used tools in such systems are Wireshark and Snort. Snort is an open-source intrusion detection and prevention system that conducts real-time packet inspection and rule-based logic to identify malicious activity. Wireshark is, nonetheless, a complex packet analyzer that monitors network traffic and offers in-depth analysis of data flow, enabling security professionals to conduct forensic analysis in depth. Such tools, if properly configured and deployed, are the foundation of an end-to-end intrusion detection and response system.

Detection is only half the picture, however. A good WNIDRS should also react to detected attacks automatically and in a timely manner. This response capability sets WNIDRS apart from conventional intrusion detection systems (IDS), which will have to act manually. In WNIDRS, if intrusion or malicious activity has been identified, the system will be able to perform preconfigured response actions such as blocking a source MAC or IP address, malicious device disconnection, hijacked access point blocking, or alerting system administrators via messages and logs. These answers would typically come under the domain of a set of rules that can be customized so that the administrators would be able to define the level of severity and respective countermeasures for various types of threats. The idea is to counter and de-escalate the threat before it becomes a full-fledged attack.

Additionally, WNIDRS provides incident logging and reporting functionality, which is essential for post-attack analysis, audit, and prevention of future attacks. Logs allow organizations to maintain a history of attempted intrusions, gain insight into attacker behavior, and refine their security policies over time. In most cases, such history can also be used to train the system's anomaly detection engine, thereby enhancing its ability to detect new threats on an ongoing basis.

As wireless technologies get better and become more ubiquitous—driven by the advent of 5G, edge computing, and the Internet of Things—the demand for wireless network security will only rise. WNIDRS in this scenario is not a response but an anticipatory, responsive, and smart security solution. It helps organizations outsmart cybercriminals through real-time visibility, rapid threat detection, and rapid response capability—all tailor-made for wireless environments.

In short, the Wireless Intrusion Detection and Response System is a critical component of modern-day network security. Its real-time detection of wireless activity, smart sensing of unknown and known threats, and automatic response to attacks make it a valuable commodity for safeguarding sensitive data and ensuring business continuity. Since computer security threats themselves evolve and wireless networks are becoming more and more the backbone of business operations, WNIDRS implementation cannot be optional but an investment that the security infrastructure of any organization cannot afford to overlook.

CHAPTER 2

LITERATURE SURVEY

"Enhanced Security: A Wireless Monitoring System with Advanced Security Features" (2023) This study by Muhammad Sufyan, Muzammil Jameel, and others, the theoretical design and implementation of an advanced wireless monitoring system were discussed that mainly focused on improving data security and intrusion protection for wireless networks. The system had several layers of security, such as encryption protocols to safeguard data in transit and authentication protocols to limit unauthorized access. One of the key features of the system was the generation of real-time alerts in cases of suspicious behavior or network anomalies. The researchers utilized wireless communication chips, such as ESP8266 and others like it, along with encryption algorithms such as AES and SHA-based authentication. The project software was accomplished mainly in Python, allowing for simplicity in deployment of warning services and secure communication channels. The architecture suggested involved sending data over encrypted channels by sensor nodes to a single monitoring point, providing confidentiality and integrity to the stream of data. One of the innovations was the balance between efficient lightweight encryption and robust security to discourage breaches. The system was validated in a controlled testbed for latency, throughput, and security robustness measurement against attack simulation. Though its promising solution, the work suffered from a lack of practical large-scale deployment in real-world settings and hardware-level testing vital for real-world reliability measurement. The work also failed to illuminate power control and long-term sustainability of the wireless nodes. Future studies suggested by the authors included integration with IoT-based infrastructure and industrial and urban surveillance application cases scaling the system's versatility.

"Performance of Kismet Wireless Intrusion Detection System on Raspberry Pi" (2022) Al-Attar and Aldabbas investigate the possibility of placing the Kismet Wireless Intrusion Detection System (WIDS) on low-cost hardware—the Raspberry Pi platform, to be precise. The need for their investigation comes from the possibility of presenting a low-priced, lightweight, and power-savvy wireless intrusion detection service for small-size networks—namely, household networks, home offices, or field monitoring purpose. The performance of

the system in real-time wireless network traffic monitoring will be analyzed based on CPU usage, memory consumption, and overall detection performance.The researchers installed an open-source network detector, sniffer, and intrusion detection system, known as Kismet, on a pair of versions of the Raspberry Pi to determine how well the hardware can perform the real-time packet analysis task and real-time monitoring task. Wireless adapters utilizing monitor mode were sniffing raw traffic, and performance statistics were being tracked under varied scenarios such as idle modes, active scanning, and simulated attacks. The research presented in-depth information about how low-power devices could be utilized to improve network security without any full-size infrastructure or costly apparatus. The standout characteristic of this research was its pragmatism, demonstrating how low-power like Raspberry Pi can enable Kismet to operate with adequate performance to perform minor network intrusion detection. The authors were successful in demonstrating that Raspberry Pi was adequate for extended monitoring without experiencing a noticeable loss in performance or overheating. This renders it a good choice for quick deployment in cost-effective environments or in the field.It also must be mentioned what various limitations the paper imposed. The deployment had been restricted to basic intrusion detection methods—effectively signature-based detection—and wasn't using higher-level methods like behavioral analysis or machine learning-powered anomaly detection. Also, the paper didn't mention anything on real-time auto-response against the threats, something that can form a critical element in sophisticated security systems. The system was somewhat reactive, making alerts without deploying countermeasures. mostly reactive, warning without deploying countermeasures.

"A Wireless Intrusion Detection System for 802.11 WPA3 Networks" (2021) Neil Dalal, Nadeem Akhtar, and Anubhav Gupta have created a native Wireless Intrusion Detection System (WIDS) specifically designed for monitoring and security of 802.11 WPA3-encrypted wireless networks as per the requirement. Watching the increasing adoption of WPA3 as a wireless security protocol in the contemporary age, the researchers tried to combat future vulnerabilities in this protocol—i.e., Denial-of-Service (DoS) attacks that might very well evade conventional defenses. Their framework was built on top of a custom packet analyzer just to record unusual traffic patterns and attack patterns against WPA3-specific handshake and authentication protocols.The test was performed in a test lab

environment with spoofed DoS attacks being executed to test the responsiveness and responsiveness of response of the system. The tools used were WPA3-set wireless routers, Linux operating systems, and Python for writing custom scripts to capture and analyze intercepted packets. The detection engine was particularly scanning for deauthentication frames, unusual retransmissions, and handshake anomalies that are usually indicators of impending DoS attacks. One of the most powerful elements of the system was prioritizing WPA3's new security model, such as the Simultaneous Authentication of Equals (SAE) handshake. This enabled the WIDS to scan traffic with an improved knowledge of protocol-level nuance, making more accurate threat detection than run-of-the-mill WIDS solutions. Another benefit was that the modular design provided room for the system to be integrated into other security products or for adding on detection rules.But WPA3-specific data within the system was a weak point. Although highly successful at the identification of targeted WPA3 network attacks, its effectiveness against mixed or legacy networks (for example, networks still making use of WPA2) is ambiguous. Also, no real-world large-scale deployment scenario was involved in the study, which is essential to conclude performance in real-to-life setups with dynamic network behaviors and traffic models.

"Design of a Snort-based IDS on the Raspberry Pi 3 Model" (2021)This paper by L. D. C. Silveira et al. describes the deployment of the Snort Intrusion Detection System on a Raspberry Pi 3 OS through the Raspbian OS with the aim of turning it into an affordable and portable network security appliance. The system was tested using several simulated attacks to assess the detection rate and resource utilization. Tests revealed that the Raspberry Pi 3 was able to run Snort and detect most common threats like port scans and brute-force attacks, which can be utilized for small networks and learning. However, under high traffic, the system was plagued by extremely poor performance characteristics like lagging detection and high CPU usage. The paper also states the limitation on relying solely on signature-based detection and limiting its functionality to detecting new or zero-day attacks. The IDS did not have adaptive or smart features such as anomaly detection or machine learning integration. Despite all these limitations, the study shows the promise of the employment of low-cost hardware to deploy basic intrusion detection and keeps suggesting that even further optimizations and enhancements will keep its utilization better in even more dynamic

networking environments.

"IDS Using Raspberry Pi and Telegram Integration" (2021) H. Wijaya and E. G. Rachma proposed a sophisticated intrusion detection system (IDS) with Telegram integration to provide real-time alertness for network intrusions. It leverages the capabilities of Raspberry Pi, a low-cost and multi-functional platform, and Telegram Bot API for real-time communication. Python-programmed, the system is easy to install and notify users at a high rate, giving users an early indication of potential security incidents. Through integration with Telegram, the system gives notification without involving complex user interfaces and gives availability through phones. The process involves monitoring of network traffic for something out of the norm, detection of potential intrusions, and real-time notification in the form of Telegram to the affected users. Even though the system offers real-time and interactive response, the weakness of the system is that it is network dependent, dependent on connectivity and network availability. For use where low network connectivity and/or high latency levels are the norm, warnings would be delayed and hence decrease the responsiveness of the system when used in critical and timing-sensitive applications. In spite of that, the system provides a useful cost-effective solution to small to medium-sized installations where immediate notification is needed but network reliability is not yet an issue.

"Indoor Intrusion Detection and Filtering System Using Raspberry Pi" (2020) by Deshmukh R. discusses a novel, low-footprint intrusion detection system for a special application to restrict undesirable QR code reading indoors. It employs the use of Raspberry Pi, which is a cost-effective and portable hardware platform, for scanning and securing areas from abusive use of QR codes, becoming more widely applied for purposes ranging from payments to data exchange. With the assistance of Python-based filtering algorithms, the system efficiently checks incoming QR codes for legitimacy to avoid possible security threats that may be injected through unauthorized scanning of malicious QR codes like phishing or malware installation. The use of Raspberry Pi in such a security device renders the system deployable at a low cost and suitable for small and medium-sized installations. Nonetheless, the system is limited to indoor environments, where the perils of scanning QR codes are larger and more manageable. The system would fall short in wireless intrusion detection on a large scale within dynamic or external environments, where other types of intrusion are potentially more

likely to occur. The system is therefore viable as an on-site measure for security issues but is not feasible in environments beyond those that are controlled indoors.

"Raspberry Pi-Based Intrusion Detection System" (2018) by B. G. Rasane, S. S. Mane analyzes the feasibility of employing Raspberry Pi as a platform to deploy an Intrusion Detection System (IDS) for small-scale network environments. Authors deployed a system that employs packet sniffing tools such as Wireshark and network monitor tools combined with rule-based platforms such as Ettercap to identify traffic anomalies as well as unknown device connections. The main aim was to suggest an inexpensive, light-weighted security solution deployable in a home network, small office environment, or even a university laboratory with ease. The system can identify certain network attacks such as ARP spoofing, MAC address flooding, and simple brute-force attacks based on rule-based analysis of filtered traffic. One of the strong foundations for deployment is the ease of deployment and low hardware resources needed, hence a viable option for low-budget users or organizations. The authors also suggest the possibility of more innovation with the inclusion of machine learning methods to further improve detection accuracy and response to pending threats. Integration into cloud-based alerting schemes for remote monitoring and centralized alert management is also being suggested. Although the prototype presents effective low-level threat detection, the prototype does not scale or possess the level of complexity to tackle advanced attacks or high-end attacks. Nevertheless, the research presented solid foundation for future studies in the field of IoT-based security and lightweight IDS design.

"Raspberry Pi-Based Model for Investigating Intrusion Evidence" (2018) by A. K. Sharma and K. B. Anoop in 2018 is a Raspberry Pi-powered model exclusively for digital forensics to acquire and store network intrusion-related evidence. The model works by monitoring the network and recording important information about any potential security breach, which can be a great tool to use in post-incident analysis. Cost-effective, small application of Raspberry Pi makes it easier to apply the model in any forensic analysis where cost or size is a matter. Data of network traffic are precisely recorded by the device in order to allow researchers to backtrack the sequence of events, detect the vector of attacks, and harvest valuable evidence that will be processed or prosecuted. However, while the system is better in capturing and preserving intrusion evidence, it suffers from the basic flaw that it is not designed for the detection of intrusions in real-time or for active defense. It is thus incapable of notifying

administrators or of blocking intrusions in real-time, limiting its application where the timing is critical. Despite this restriction, the system would find its application as a strong instrument of post-incident forensic analysis for determining the way an attack had taken place and providing the requisite evidence to manage security incidents.

"RaspyAir: Self-Monitoring System for Wireless Intrusion Detection using Raspberry Pi" (2017) by J. V. Divya and N. Niranjana introduces an innovative method for wireless intrusion detection through a Raspberry Pi. RaspyAir system introduces the concept of monitoring anomalies of wireless signals in terms of observing changes and irregularities of network signals. By executing Python scripts on the Raspberry Pi, RaspyAir monitors and identifies strange signal patterns that can indicate possible security intrusions, e.g., malicious use or unauthorized use of the wireless network. When detected by the system, suspicious behavior flags issues to users so that proper intervention can be made in a timely fashion. Through this self-auditing method, intrusion detection is afforded by an uncomplicated yet powerful technique that makes no demands upon costly special apparatus. But the system possesses certain good limitations. Its scalability is one such first-order problem, for it will be insufficient for increasing networks or sophisticated environments, where more than one access point and larger traffic amounts have the potential to overtax the ability of the system. Additionally, although the system is efficient in the aspect of identifying individual wireless intrusions, it fails to offer total protection for other classes of attacks like denial-of-service or even more complex exploits. Therefore, RaspyAir offers a low-profile but easy utility in scanning wireless network but does not do much better when addressing larger classes of security issues or larger-sized installations.

"Smart WIDS Implementation for SOHO Environment" (2016) by Andriansyah R. and Setiawan A. in 2016 is a Wireless Intrusion Detection System (WIDS) implemented for small office/home office (SOHO) environments. The system is implemented on the Raspberry Pi 2 platform based on Snort, an open network intrusion detection system, and has a web-based interface made for easy management and monitoring. The system is meant to fill the security needs of small-scale installations, where an enterprise-level intrusion detection system would be too heavy and expensive. The system detects and responds to suspicious behavior in real-time, and it sends timely alerts to users every time an implied security threat is identified. As much as the system has advantages, it also

brings along some real challenges. The most problematic of issues is the false positive ratio, which causes repeated alarms and make its application on customers a complexity. For a small network with dynamic network traffic, literally where demultiplexing of traffic as usual and threat traffic is an issue, it's terrifying. Besides, the system lacks centralized threat management capabilities and hence an integrated platform to manage multiple devices or networks in larger or more scattered environments. This capability holds back the system from scaling or handling advanced security threats from multiple directions. Though it is a response to limited-scale threats, its specificity and scalability would have to be raised to a level where it can contend with an expansive list of threats.

CHAPTER 3

SYSTEM REQUIREMNETS

3.1 Software Requirements

Operating System:

- KALI Linux

Libraries and Languages:

- Kismet: Kismet is a powerful wireless network detector, sniffer, and intrusion detection system for Wi-Fi networks.
- Bash scripting: used for command-line interaction and scripting in Unix/Linux systems.

3.2 Hardware Specifications

- CPU: Utilize a quad-core CPU like Intel i5/i7 or AMD Ryzen 5/7 to develop the Flutter application and execute migrations with Truffle / Broadcom BCM2712 processor, which is a 64-bit quad-core ARM Cortex-A76 CPU running at 2.4 GHz (raspberry pi).
- RAM: At least 4 GB is required. 8 GB is preferable to run kismet and simulate attacks.
- Storage: You need a 256 GB SSD/ HDD to store project files and at least 16 GB micro SD card (preferably 32 GB) to store KALI Linux OS in raspberry pi.
- Network: A stable internet connection with 10 Mbps upload/download and an external Wi-Fi adapter with Atheros AR9271 chipset to monitor networks.

3.3 Methodology & Data Set

Stages:

1. Requirement Analysis

- Define security requirements (e.g., rogue client detection, DoS attacks, unauthorized access points).
- Assess wireless environment (e.g., number of users/devices, area, size).

- Insert network design and sensor deployment steps here.

2. Network Design and Sensor Deployment

- Make sensor location decision:
- Put WIDS sensors in a manner that all areas covered by wireless are covered.
- Set minimum overlap to avoid redundancy.
- Choose right hardware:
- Single-function sensors or dual-function WIDS-enabled access points.

3. Data Collection

- Gather data through wireless traffic monitoring with sensors:
- Capture packet on channels (802.11 a/b/g/n/ac).
- Gather metadata: SSIDs, MAC addresses, signal strength, channels, etc.

4. Intrusion Detection Mechanisms

- Signature-based Detection:
- Detect known attack patterns (e.g., MAC spoofing, de-authentication floods).
- Anomaly-based Detection:
- Build baselines of normal behavior and report deviations.
- Policy-based Detection:Build rules (e.g., "Only SSIDs with prefix 'Corp-' allowed").

5. Threat Categorization and Alerting

- Classify threats: Rogue APs, Evil Twin, DoS, Man-in-the-Middle (MITM), MAC spoofing, etc.
- Trigger alerts:Real-time administrator notifications.
- Log suspicious events for later analysis.

6. Response Mechanism

- Automated response :Disable malicious devices,lock malicious MAC addresses
- Manual response:Security team analyze and respond.

7. Logging and Reporting

- Maintain records of all events that are detected.
- Produce regular reports:
- Attack patterns.
- Device behavior patterns.
- Coverage effectiveness.

8. System Maintenance and Updates

- Signature-based Detection:
- Detect known attack patterns (e.g., MAC spoofing, de-authentication floods).
- Update detection signatures regularly.
- Tune anomaly detection thresholds to reduce false positives.
- Audit and review WIDS policies regularly.

CHAPTER 4

PROPOSED APPROACH , MODULES DESCRIPTION, AND UML DIAGRAMS

4.1 Proposed Approach

Current trends in technology have shown how wireless networks are susceptible to hacking as a result of its accessibility. Furthermore, community wireless networks are usually targeted because of their lack of security. To counter this vulnerability, Wireless Intrusion Detection and Response System (WIDRS) provides a proactive security solution that is real-time. WIDRS has been tested on minimal hardware such as the Raspberry Pi or Ubuntu based systems and the results show that it can detect and respond to numerous wireless threats.

The system attempts to prevent De-authentication Floods, ARP Spoofing, Evil Twin, Denial of Service attacks, and a host of others. WIDRS uses standardized software such as Aircrack-ng, tcpdump, Snort, iptables, etc. to monitor the wireless traffic. After collecting the required data, the system implements various techniques to identify whether the activity is abnormal. Once abnormal activity is detected, the system responds by automatically blocking the attacker's IP, notifying other users, and deleting unauthorized users connected to the network.

WIDRS employs modular architecture which means that monitoring, detection, response, and logging are different modules. This arrangement allows for easier management of the system, flexibility when it comes to updates, and scalability for various environments.

System Architecture:

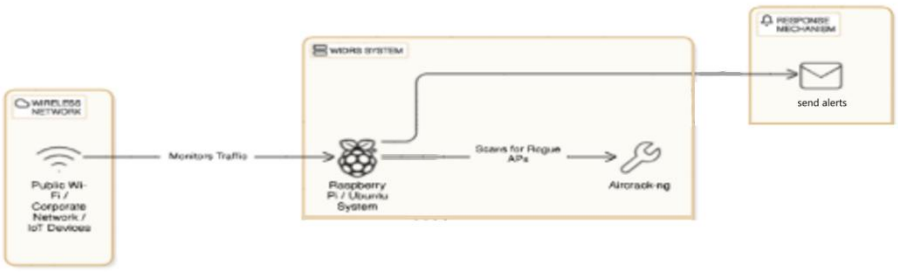


Fig No 4.1.1 System Architecture

The Fig. 4.1.1 shows a Wireless Intrusion Detection and Response System (WIDRS) which is designed to scan and protect wireless networks such as public Wi-Fi, company network, and network of IoT devices. The core of this system is a Raspberry Pi that continues to monitor traffic on the network and handles security functions in general.A system of WIDRS performs multiple things like scanning rogue access points using Aircrack-ng, managing firewall rules with iptables. In addition to these, it utilizes Snort for intrusion detection and tcpdump for traffic capture and inspection. In the event of suspicious activity, it initiates a response mechanism in which it sends email notifications and Telegram alerts for notification of administrators. It also mitigates threats by utilizing mitigation techniques such as the prevention of deauthentication attacks and blocking of IP to prevent attacks. This architecture provides a low-cost, proactive wireless network security and real-time intrusion response solution.

4.2 Modules Description

1. System Configuration and Management Module

Purpose:

To facilitate initial setup, customization, and ongoing management of WIDRS system modules and configurations.

Key Features:

- Service management for starting, stopping, and restarting modules.
- Performance tracking including CPU, memory, and network usage.
- Configuration wizard for system initialization and interface selection.

2. Network Monitoring Module

Purpose:

Ongoing monitoring and scan for wireless network traffic to identify abnormal behavior or unauthorized access attempts.

Key Features:

- Capture packets with Aircrack-NG.
- All network traffic is run in mode operated to sniff.
- Channel supports hoping to find hidden or illegal access points.
- Maintain traffic logs to monitor and check.

3. Intrusion Detection Module

Purpose:

Analysis of intercepted traffic and detection of malicious activity according to the signature and user-defined rules of the known attack

Key Features:

- General attacks like deauth, evil twin and fake AP spamming.
- User can specify their own detection rules
- Suspicious activity gives real-time alert as soon as it is detected.

4. Dashboard and Logging Module

Purpose:

Preserve historical scraps logs and in the future monitor alerts, keep central point of hands for administrators control system performance.

Key Features:

- Keep protecting access by covered logged by authorized individual.
- Alter dialect too to make description appropriate and self speaking.
- Redo algorithms to accomplish accurate tracking like group stats over networks in no-alert triggered intervals.
- Network Monitoring and System Health provides general over view, and obtains excess log filter check to support most constrain check table.

4.2 SYSTEM DESIGN

UML DIAGRAMS

4.2.1 **Use Case Diagram:** This use case diagram shows the functionality of a Wireless Intrusion Detection and Response System (WIDRS), which observes and processes network traffic in order to recognize anomalies initiated by an attacker. When it has recognized a threat, the system takes action to respond by alerting the admin and automatically remediating the attack, i.e., by blocking the IP of the attacker or by sending deauthentication packets. The admin, once alerted, may analyze the threat and respond further if necessary, thus ensuring a proactive and effective defense mechanism against wireless network threats.

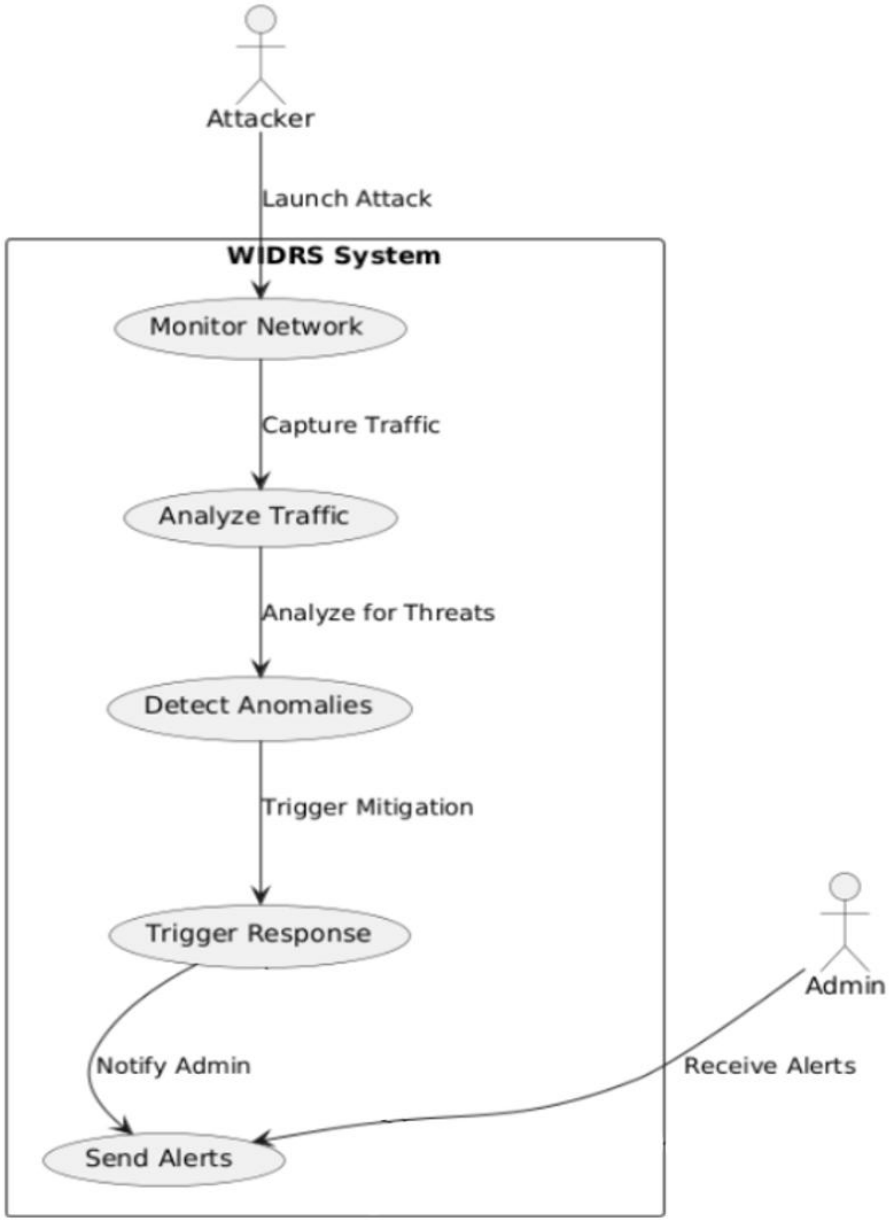


Fig No 4. 2.**1 Use Case Diagram**

4.2.2 **Class Diagram:** The class diagram is used to display the Wireless Intrusion Detection and Response System (WIDRS) structure and show interaction among its most important components. The WIDRS class plays the role of the controller and will be responsible for network monitoring, traffic analysis, and triggering responses. It uses the WirelessNetwork class for sniffing traffic and the DetectionTools class to allow traffic analysis. When a threat is detected, the ResponseMechanism class neutralizes by blocking the attacker or warning. The Admin class is alerted and may examine and respond to them for human intervention. The Attacker class is the source of threats, which launch attacks triggering the detection and response mechanism in the system.

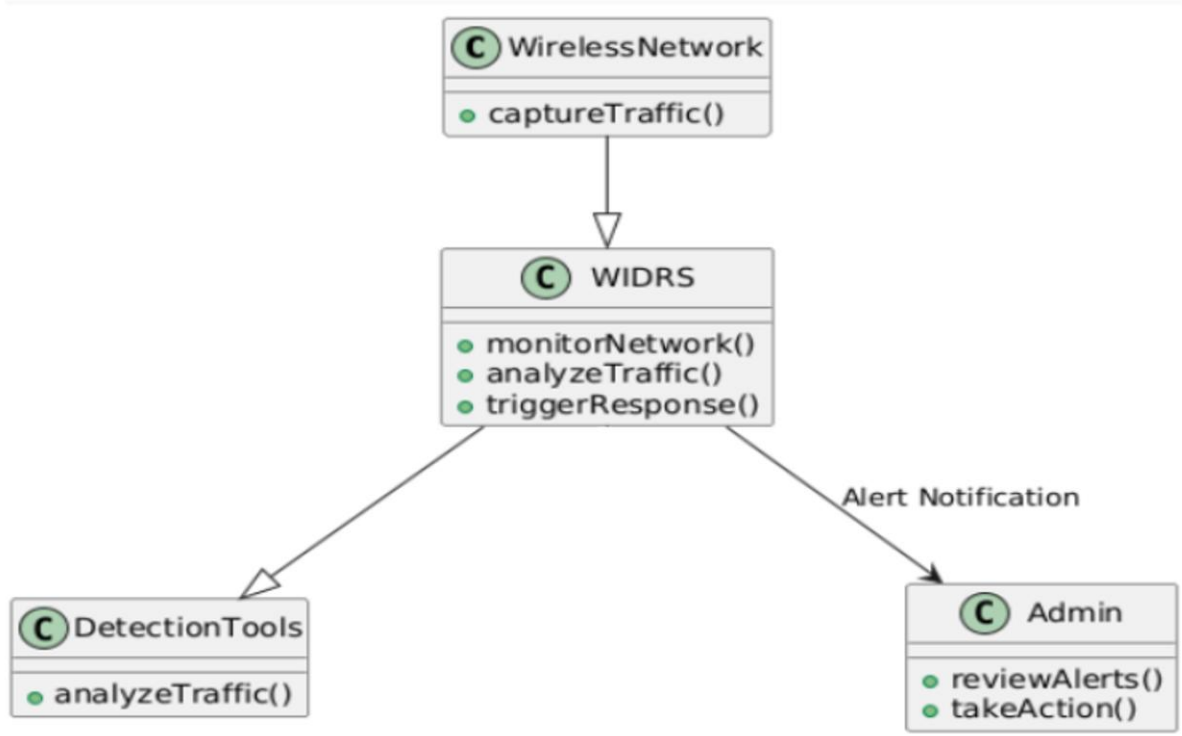


Fig No 4.2.2 **Class Diagram**

4.2.3 **Sequence Diagram:** The below sequence diagram shows the coordination among the different components of a Wireless Intrusion Detection and Response System (WIDRS) when it responds to a security attack. The process begins when the attacker sends an attack, for example, deauthentication or ARP spoofing. The wireless network captures the malicious traffic and sends it to the WIDRS system. The system scans the traffic using tools such as Aircrack-ng, Snort, or tcpdump and identifies anomalies. When a threat is identified, the system alerts it and enables mitigation measures, like IP blocking or deauthentication, through the response mechanism. At the same time, an alert is generated to the admin through email, Telegram, or system logs. The admin analyzes these alerts and takes necessary action to secure the network. This diagram is obviously illustrating the real-time data exchange and response towards wireless network attacks.

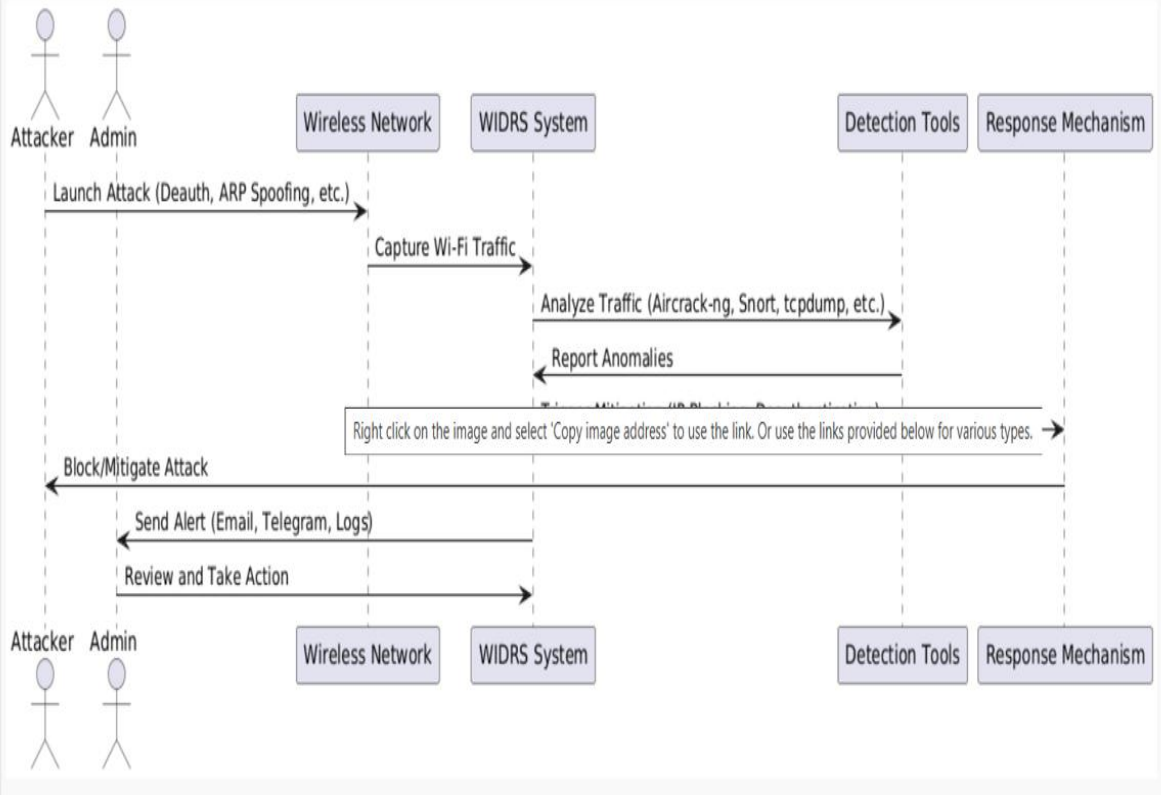


Fig No 4.2.3 **Sequence Diagram**

CHAPTER 5
IMPLEMENTATION, EXPERIMENTAL RESULTS &TEST CASES

5.1 Implementation:

Monitoring Device(RaspberryPi):

update and upgrade of system:

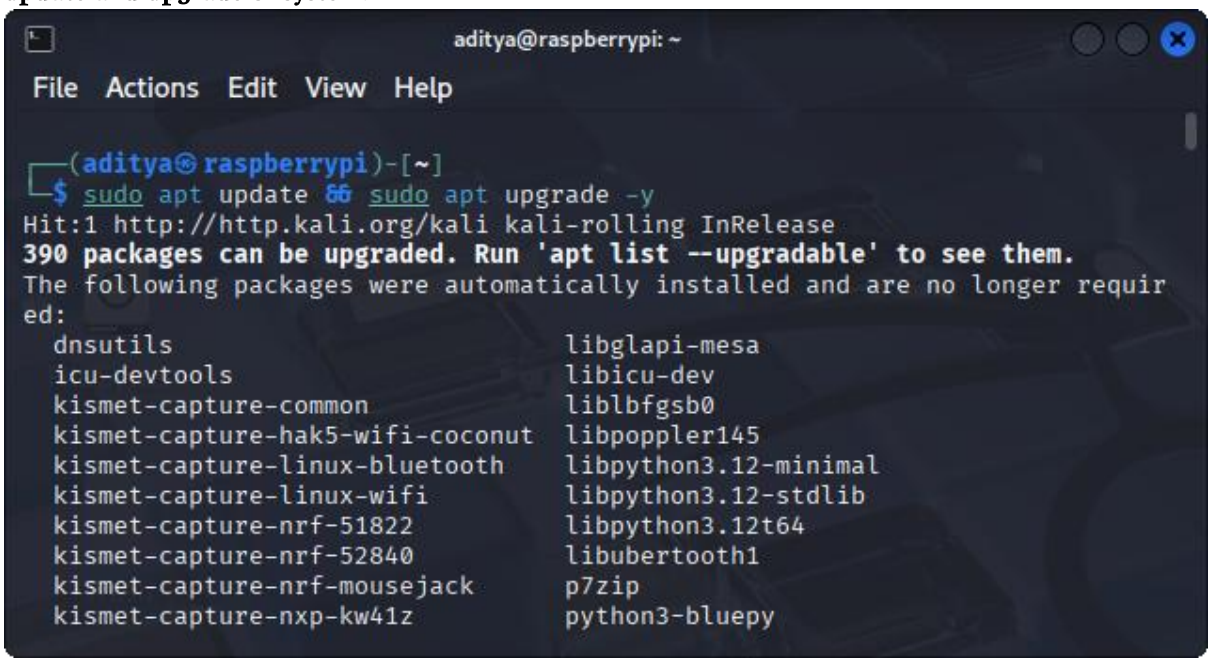


Fig 5.1.1 **update and upgrade**

Kismet tool Installation:

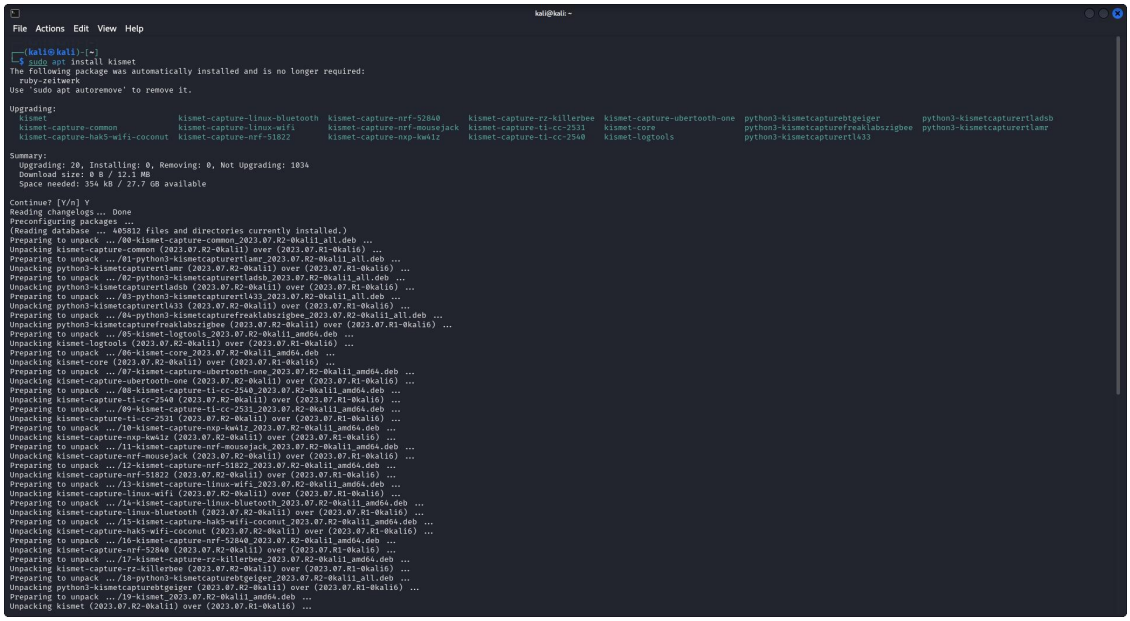


Fig 5.1.2 **kismet installation**

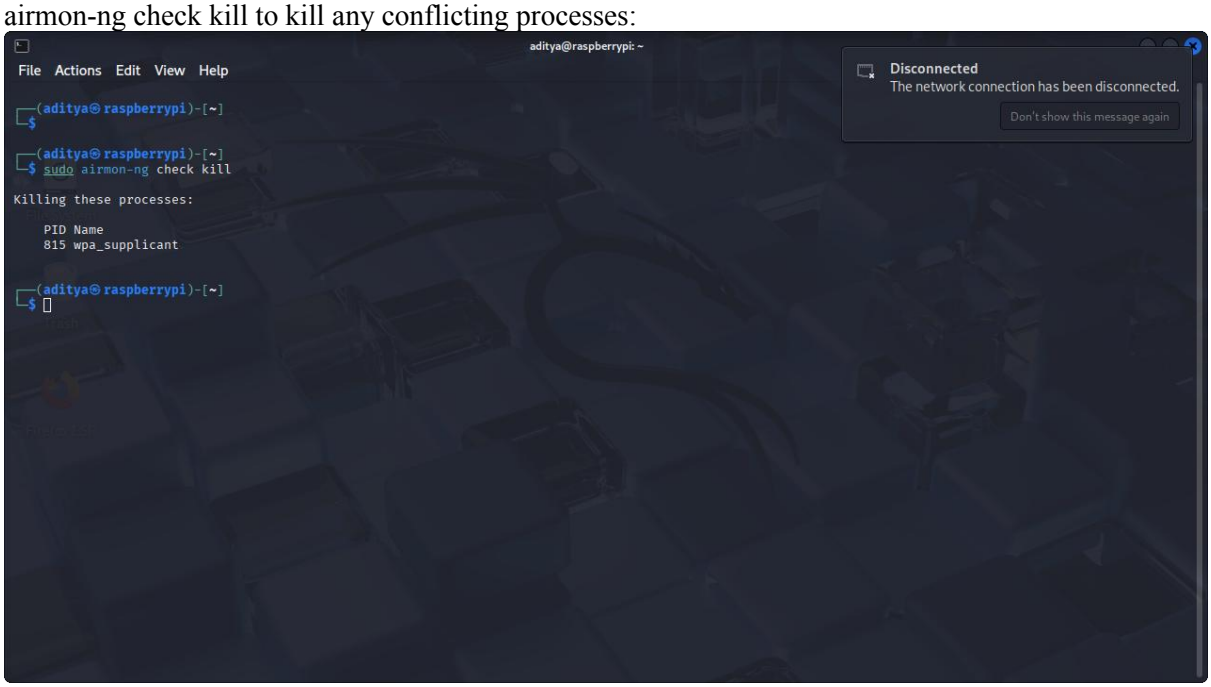
iwconfig before setting adapter to monitor mode:

Fig 5.1.3 iwconfig before mon

airmon-ng to set adapter to monitor mode:

Fig 5.1.4 **airmon-ng**

airmon-ng check kill to kill any conflicting processes:



iwconfig after setting adapter to monitor mode:

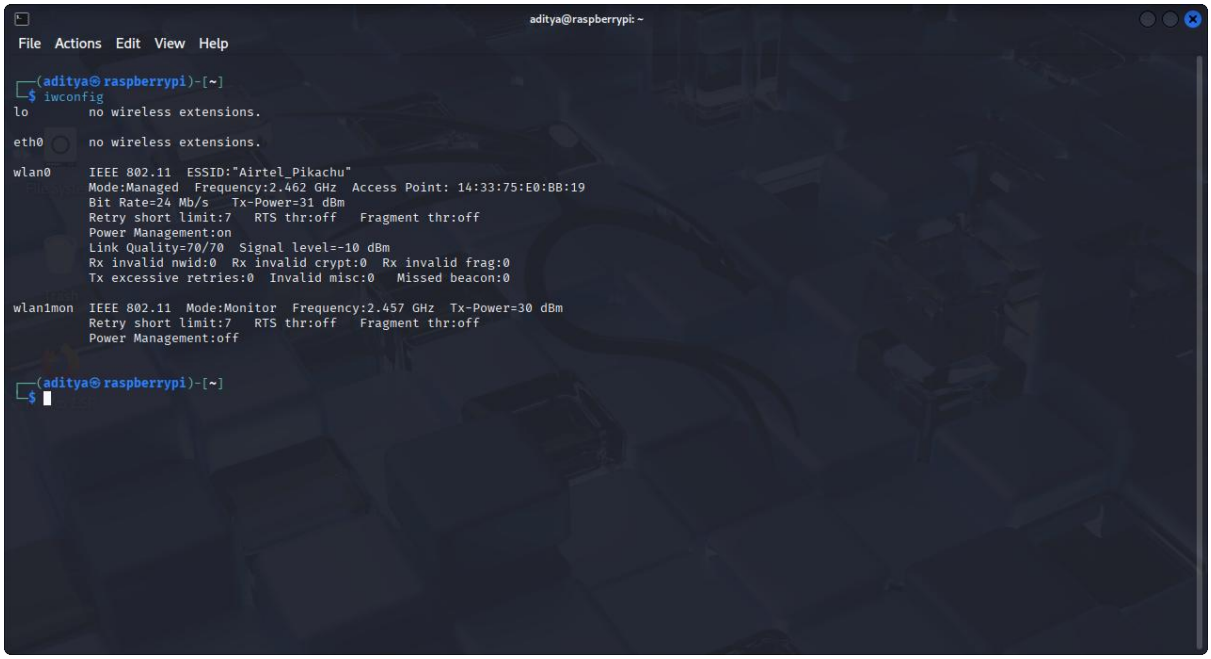
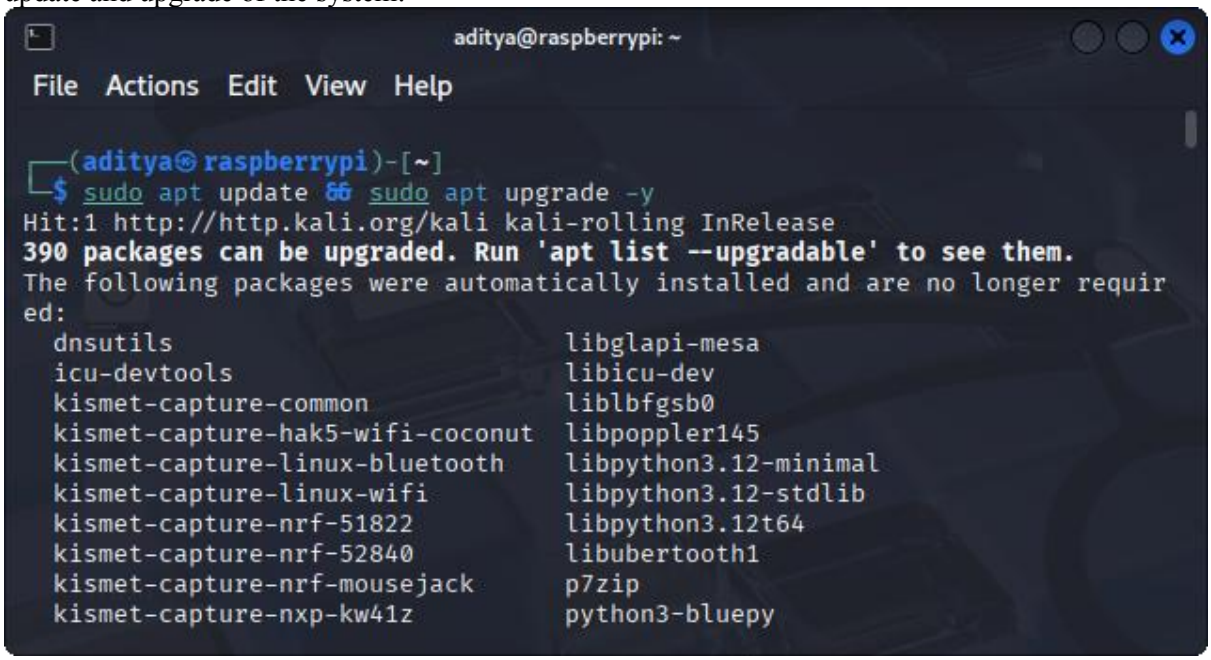


Fig 5.1.5 **iwconfig after mon**

Attacking Device:
update and upgrade of the system:



install mdk4
tool

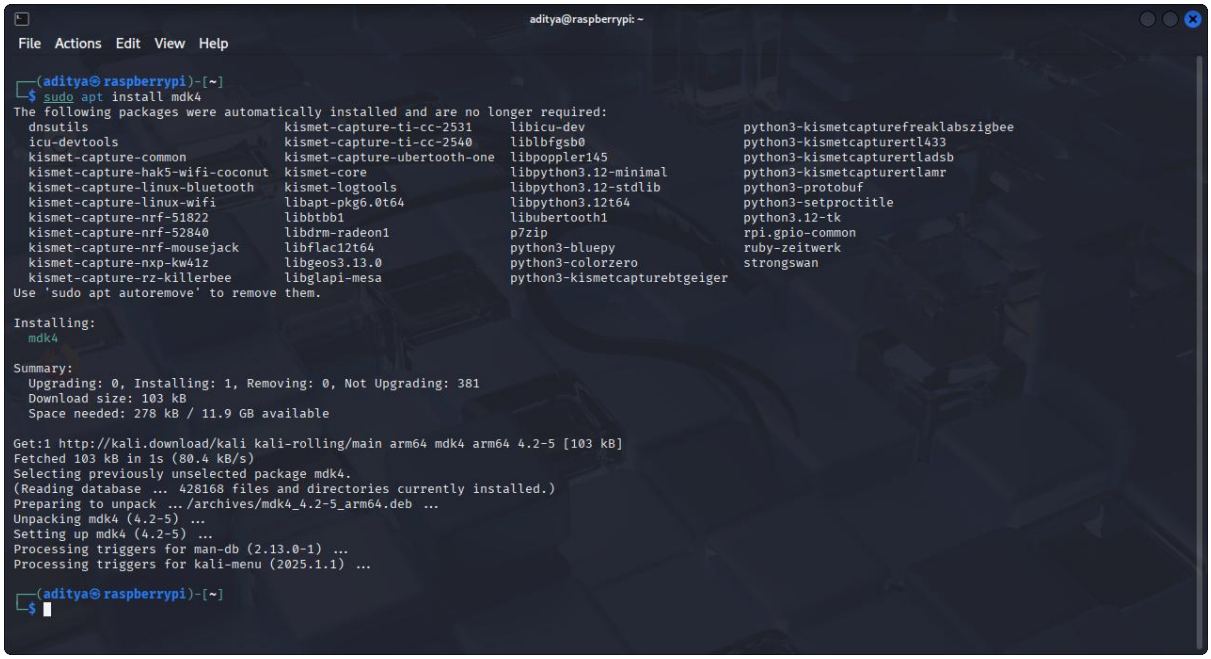
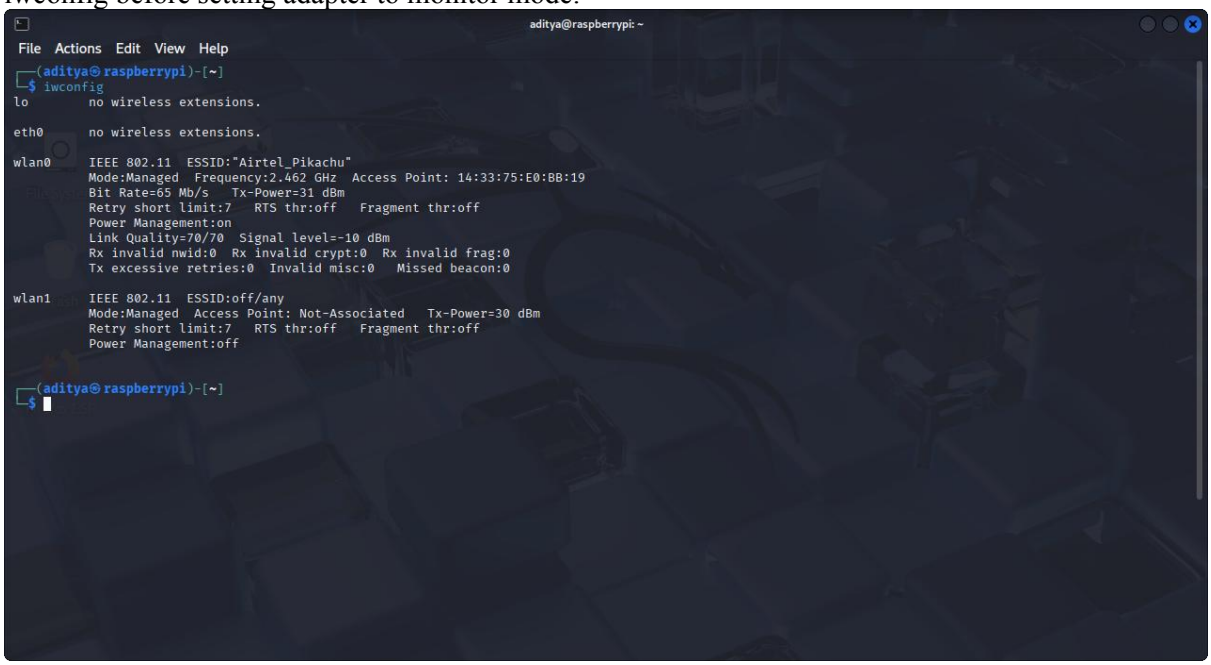
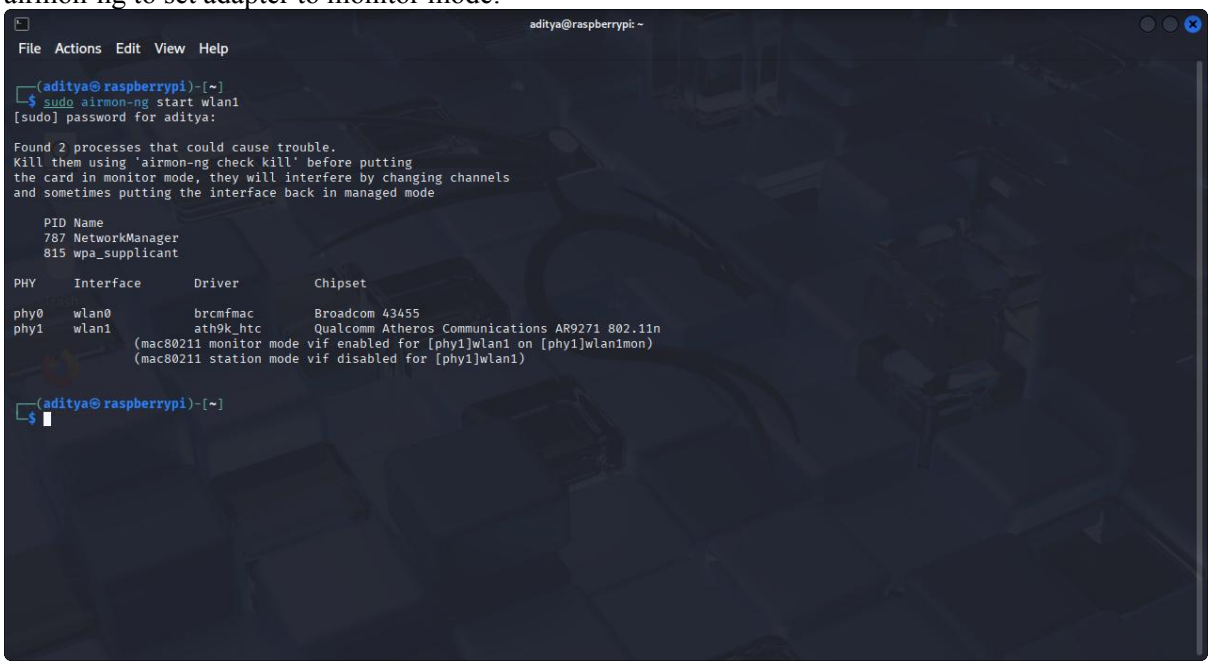


Fig 5.1.6 install mdk

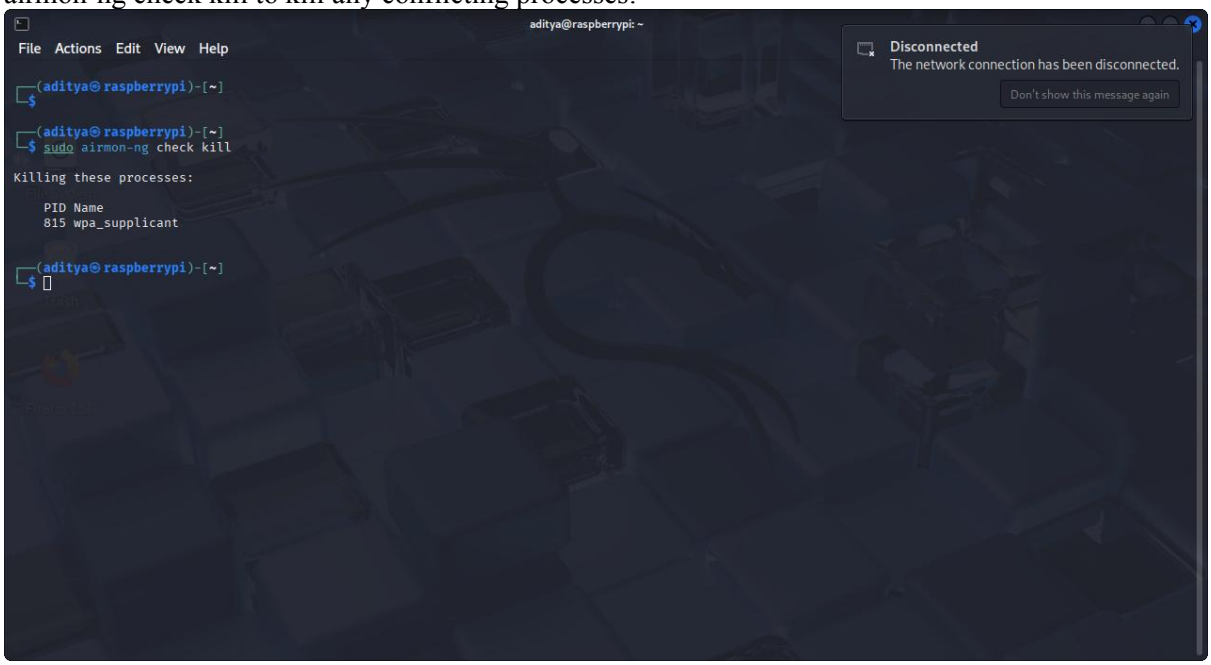
iwconfig before setting adapter to monitor mode:



airmon-ng to set adapter to monitor mode:



airmon-ng check kill to kill any conflicting processes:



iwconfig after setting adapter to monitor mode:

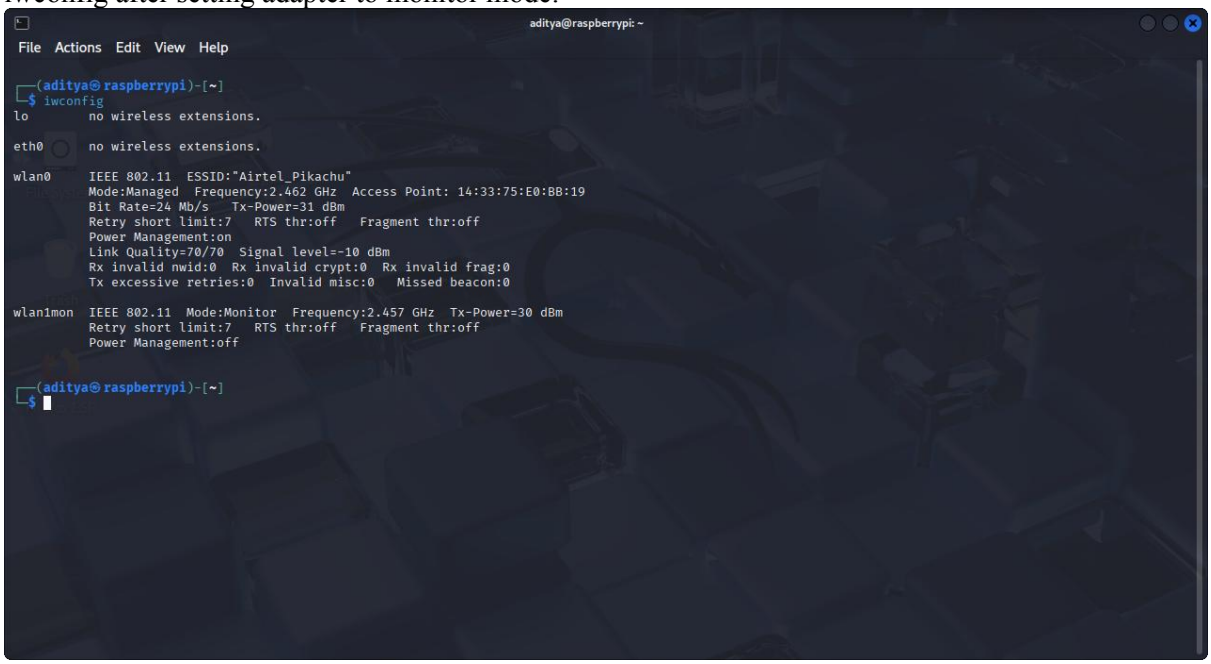


Fig 5.1.7 **iwconfig after mon**

airodump-ng , to find the MAC address of the target network:

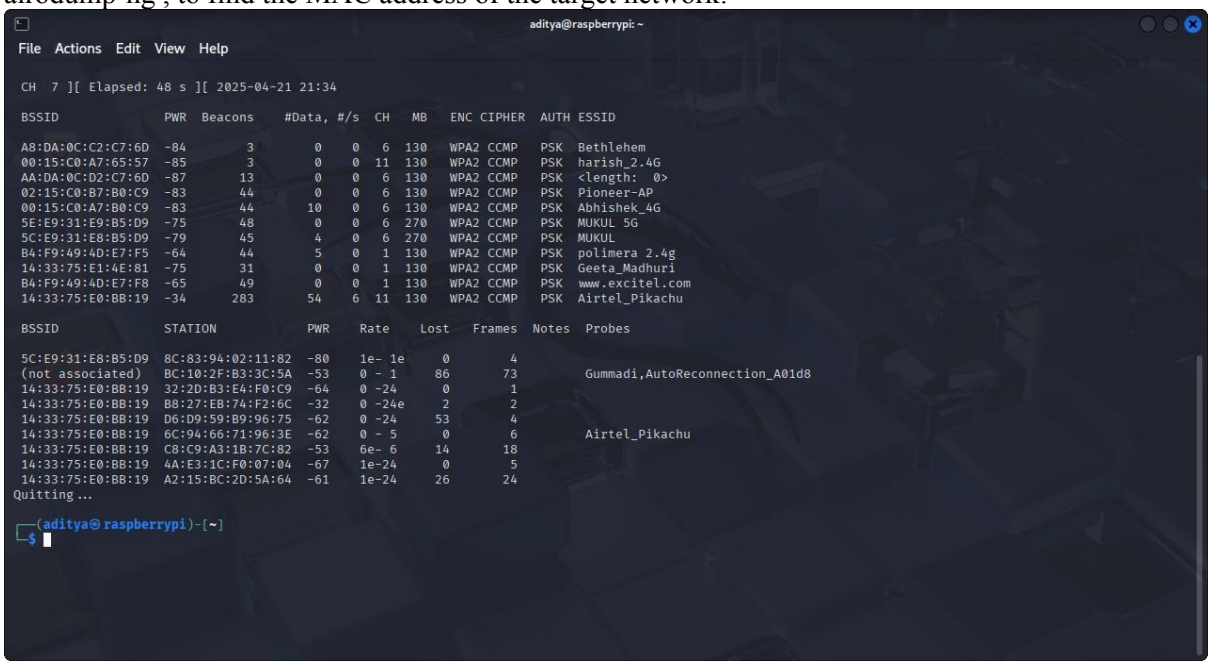


Fig 5.1.8 airdump-ng

set channel of attacker adapter to the channel of victim network:

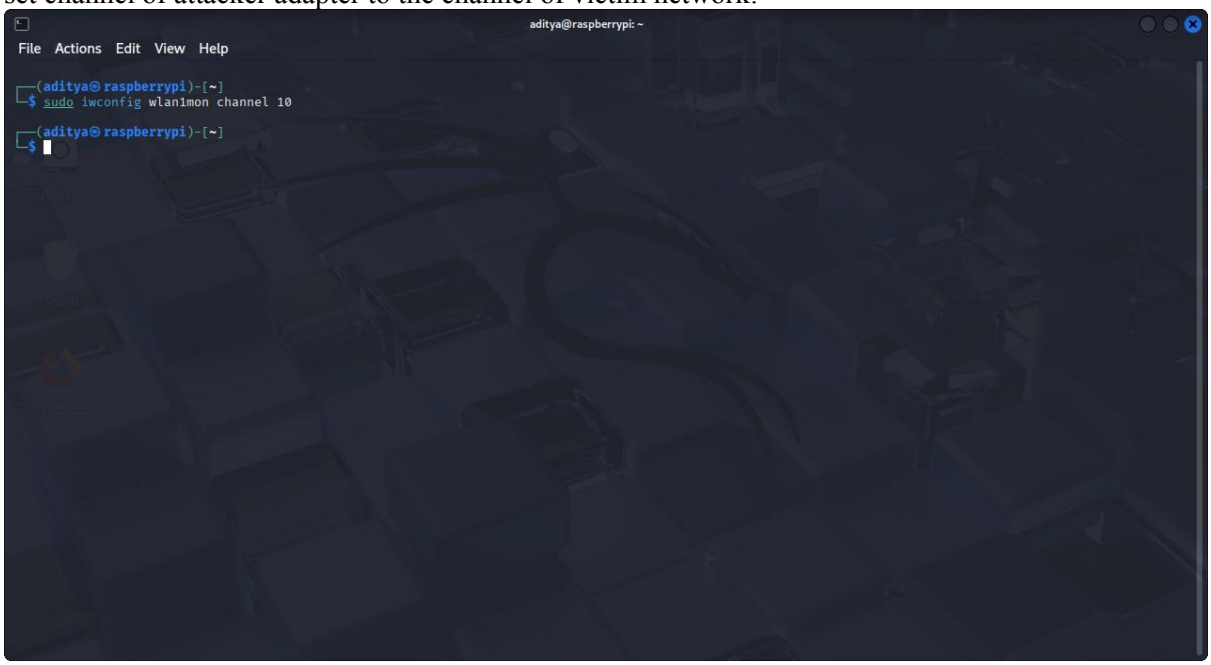


Fig 5.1.9 **set channel**

start kismet detection:

Fig 5.1.10 **kismet**

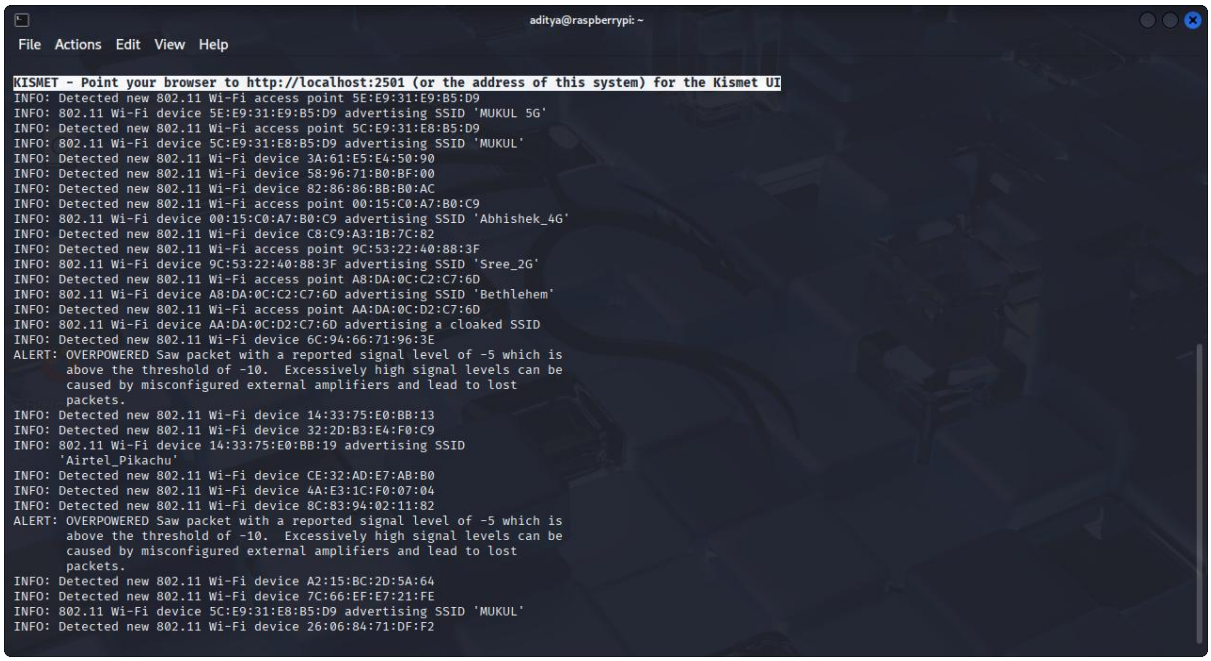


Fig 5.1.11 **kismet dashboard**

kismet-dashboard:

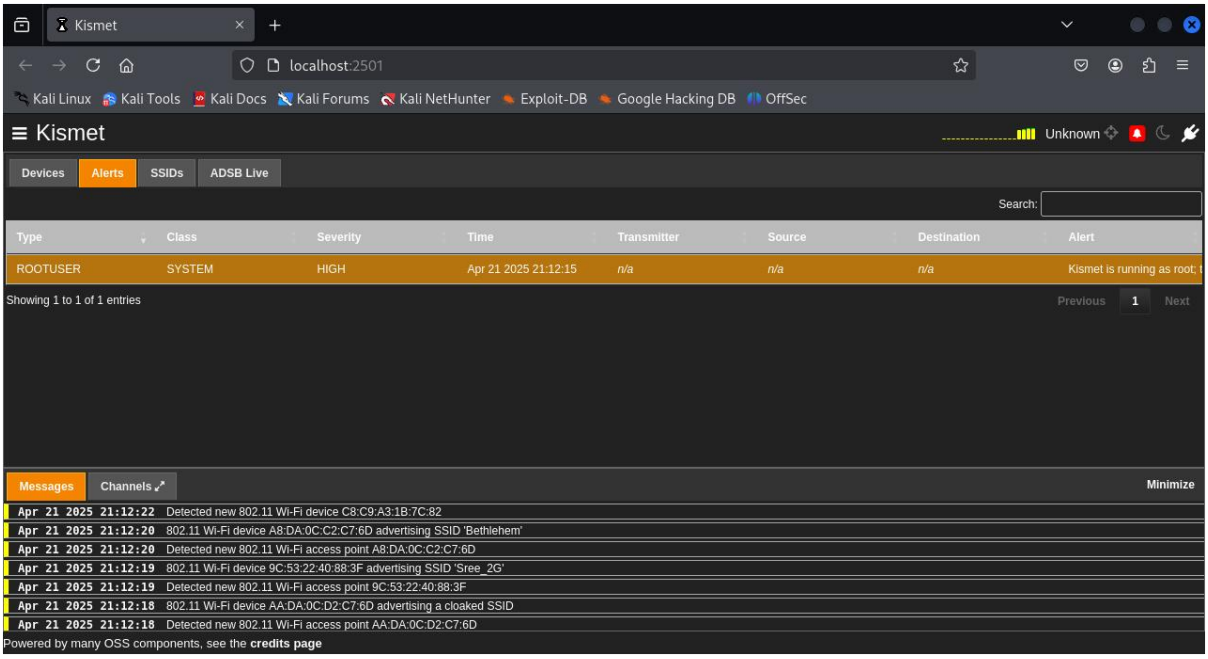


Fig 5.1.12 kismet dashboard

deauthentication attack command**:**

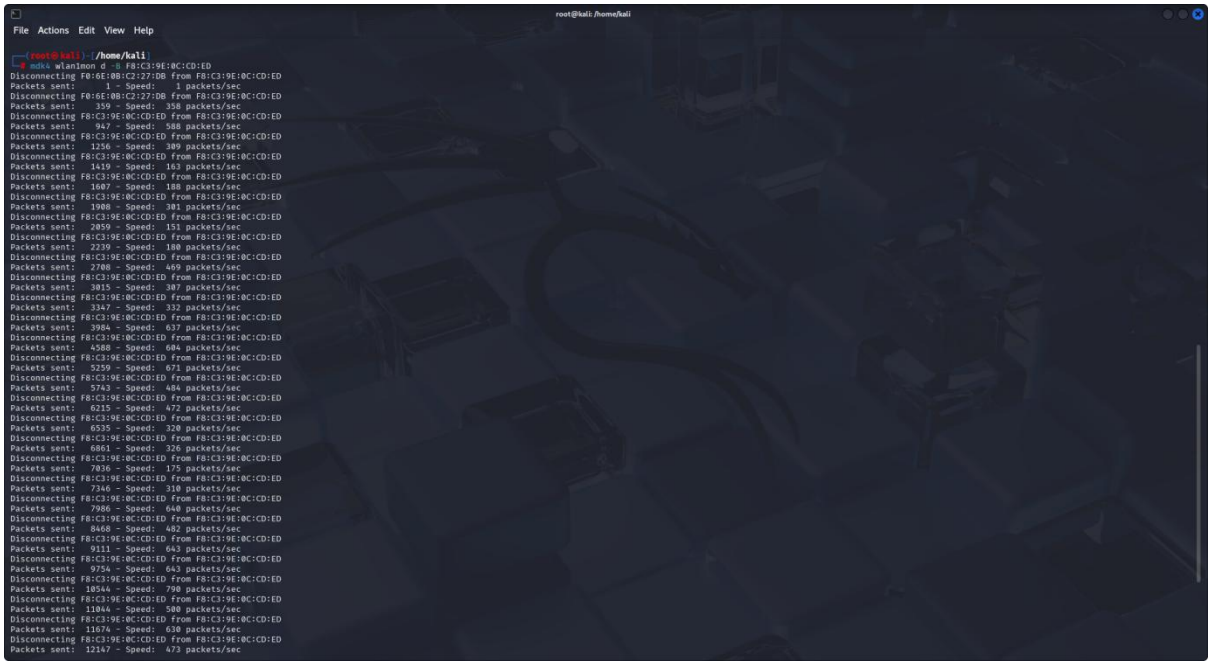


fig 5.1.13 **kismet dashboard**

fake access point flood attack command:

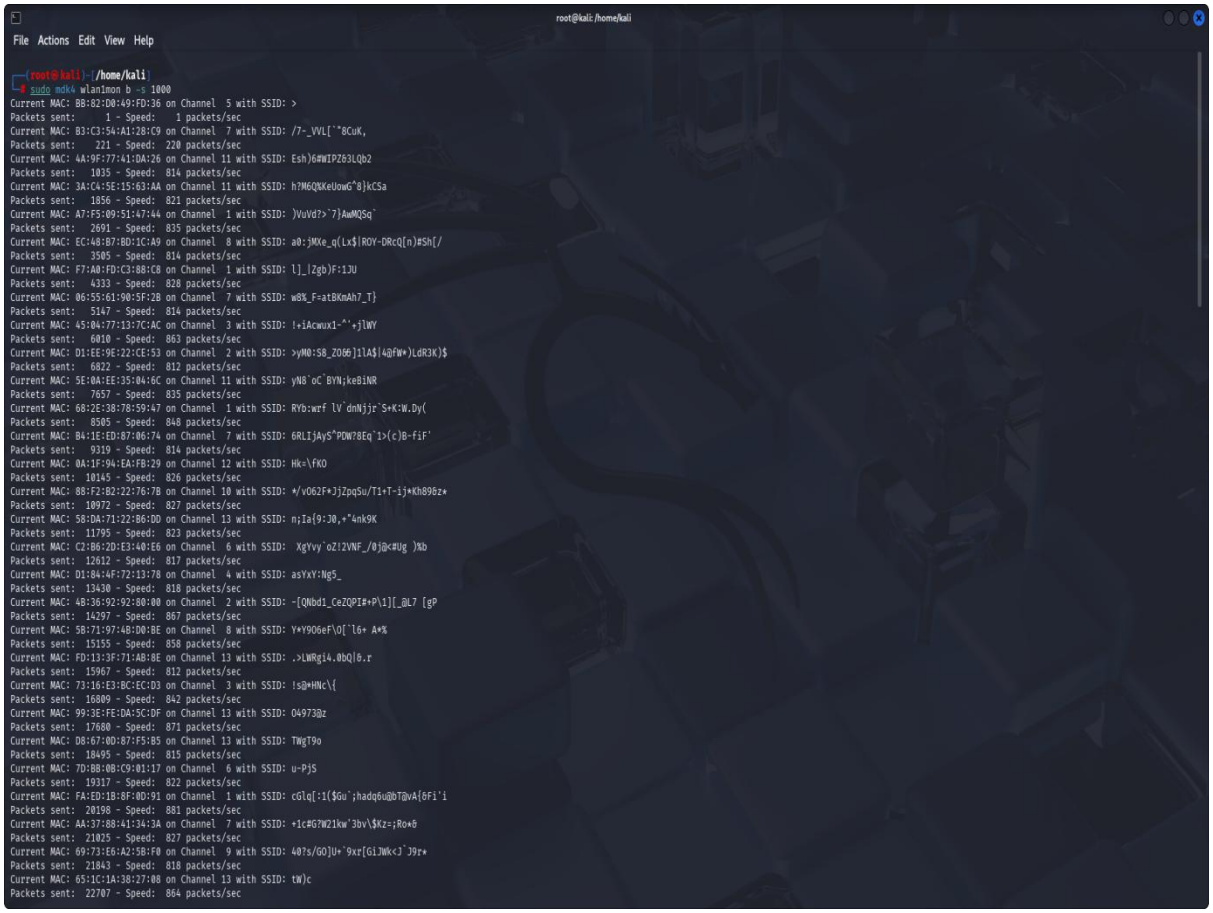


fig 5.1.14 kismet dashboard

Evil-Twin attack command:

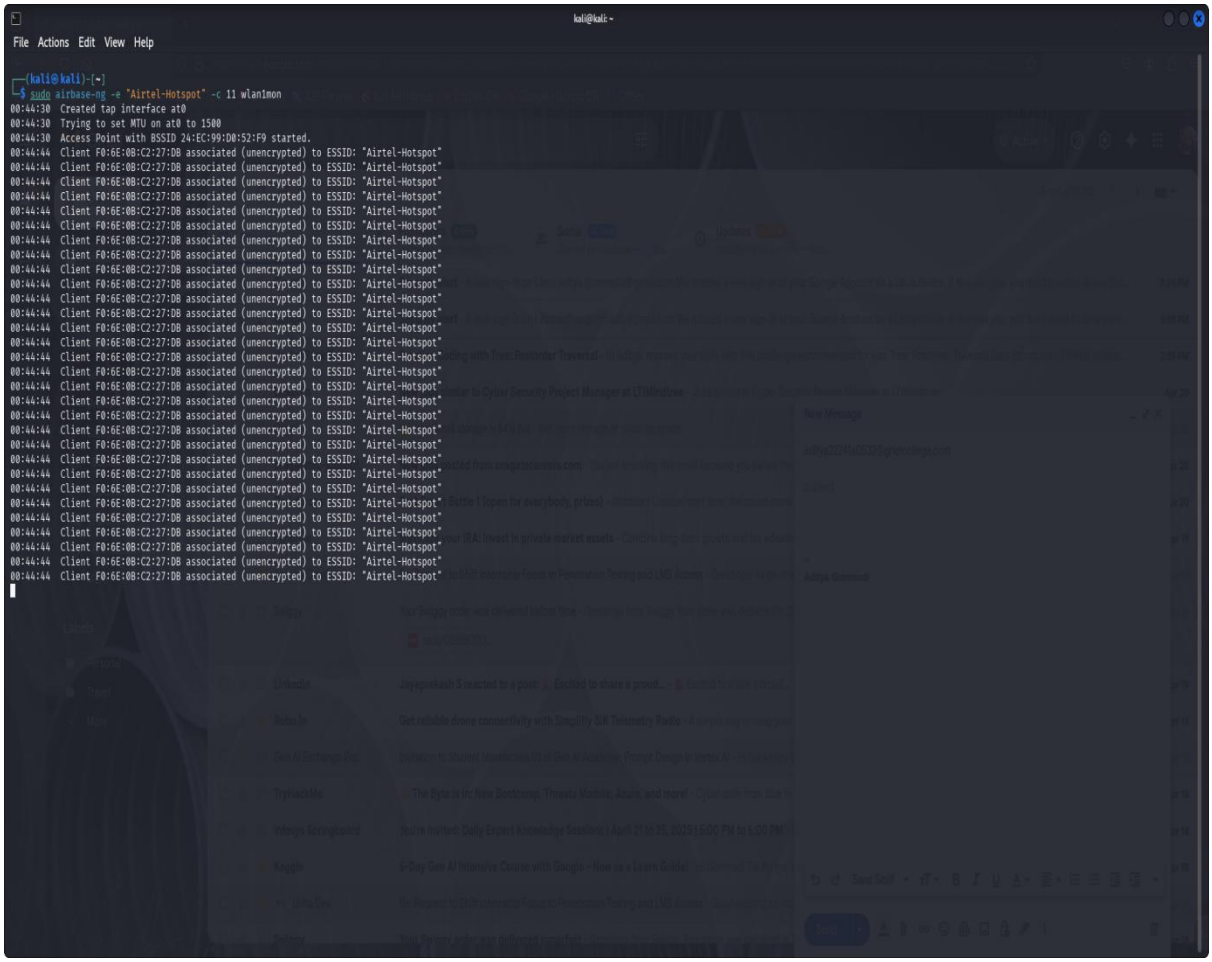


fig 5.1.15 kismet dashboa

5.2 EXPERIMENTAL RESULTS:

deauth-dashboard:

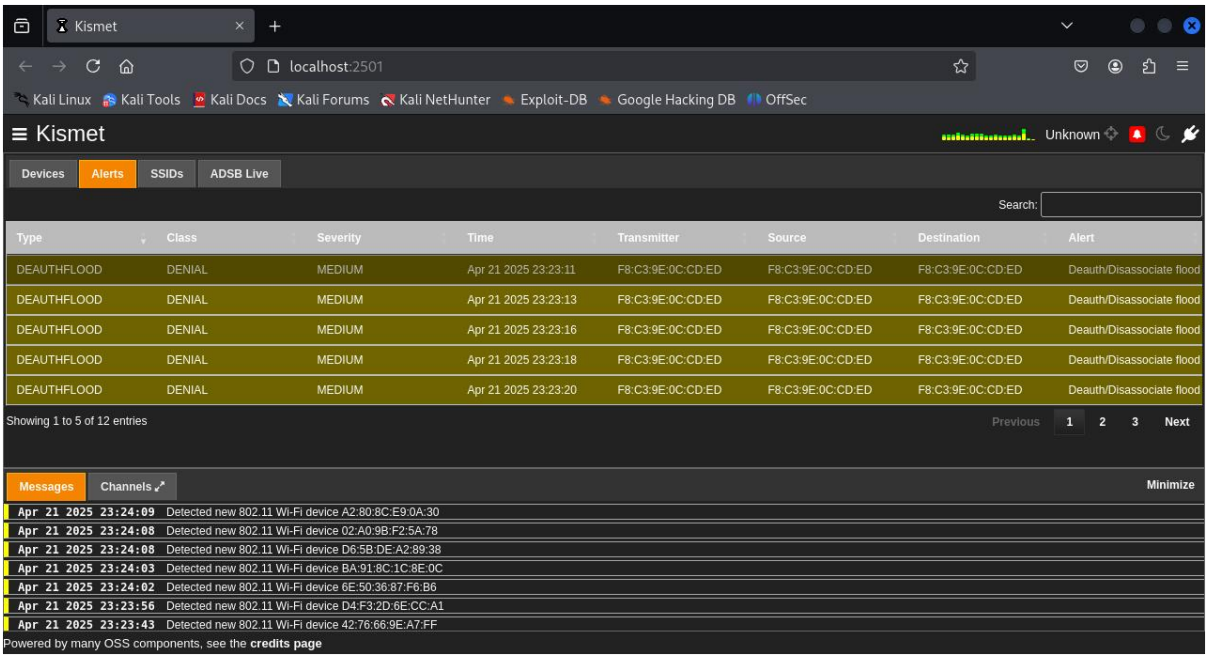


Fig 5.2.1 deauth-dashboard

Fig: 5.2.1 illustrates the Kismet wireless network detector and intrusion detection system (IDS) interface with the "Alerts" tab. The alert table contains several instances of a "DEAUTH/FLOOD" type attack under the category "DENIAL" with medium severity. Each row shows the timestamp, transmitter MAC address, source and destination, and the type of alert (Deauth/Disassociate flood) meaning a denial-of-service attempt on wireless networks. Under the alerts area, the "Messages" tab records sightings of new 802.11 Wi-Fi devices and their MAC addresses and sighting timestamps, helping monitor wireless networks in real time and detect threats.

fakeap-dashboard:

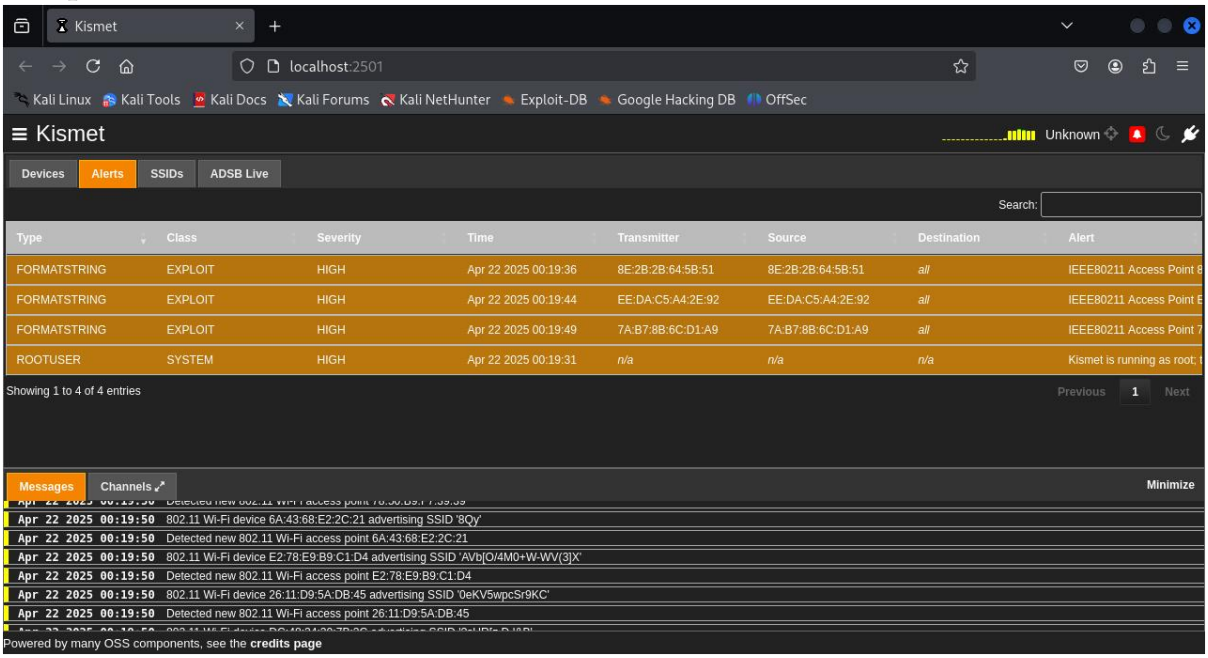


Fig 5.2.2 fakeap-dashboard

Fig: 5.2.2 indicates the Kismet interface within the "Alerts" tab, which identifies several high-severity intrusion detection events. The alert types designated that are present include "FORMATSTRING" as an "EXPLOIT" and "ROOTUSER" as "SYSTEM". These two alerts indicate the format string vulnerability and root user access detection, both of which are serious network security threats. Each record holds the time stamp, transmitter' and source' MAC addresses, and distinct alert type such as "IEEE80211 Access Point Format String." The bottom message log tracks new 802.11 Wi-Fi device captures with MAC addresses and SSIDs, providing real-time awareness of local wireless activity and potential security anomalies.

eviltwin-dashboard:

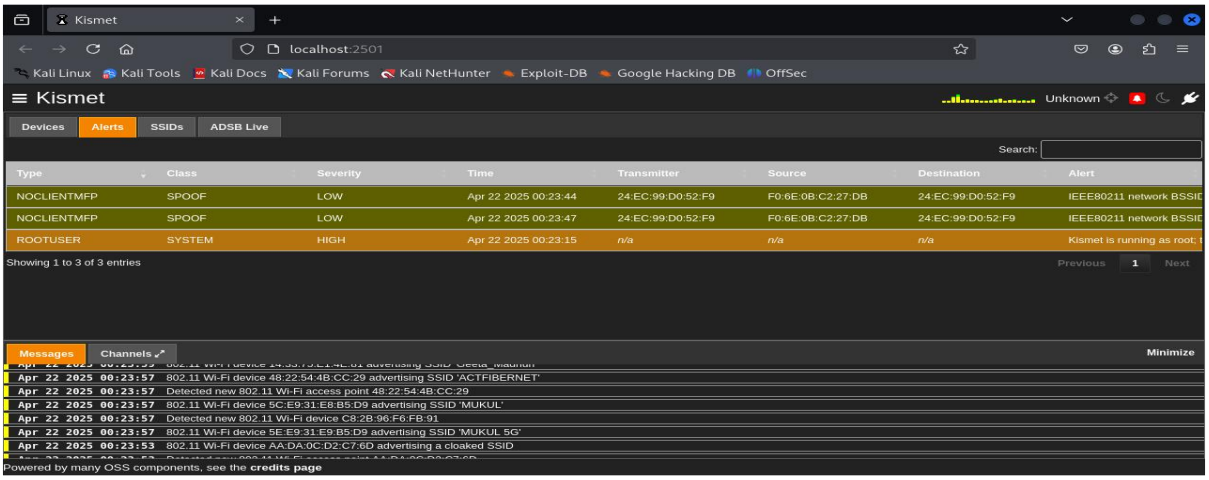


Fig 5.2.3 eviltwin-dashboard

Fig: 5.2.3 is the "Alerts" page from the Kismet wireless network intrusion detection system and shows alerts of system and spoofing attacks. Two are those which belong to the "SPOOF" class of "LOW" severity but uniquely mark "NOCLIENTMP" alerts that signify the devices that might be spoofing the legit clients within the network. These are alerts that name the transmitter MAC address and the source MAC addresses being spoofed, with "IEEE80211 network BSSID spoof" the particular problem. There is also a "ROOTUSER" alert of "HIGH" severity that indicates Kismet is running in root user mode, something of importance in engaging in sophisticated monitoring operations. Under this, the "Messages" category logs 802.11 Wi-Fi devices of various types appearing, as well as promoted SSIDs, towards active network usage detection and potential intrusion.

Test cases:

Table: Attack Detection Testing

Test Case No.	Description	Status	Remarks	Correction
1.	Detect Deauthentication Flood Attack	Success	Kismet detects and logs DEAUTHFLOOD alert .	-
2.	Detect Evil Twin (Rogue AP) Attack	Failure	No alert or EVILAP alert shown when duplicate SSID is broadcasted	Improve SSID/BSSID conflict detection logic
3.	Detect Fake AP Flood (Format String Attack)	Success	High severity FORMATSTRINGalerts triggered	-

Table: 5.3.1 Attack Detection Testing

Table: Integration Testing with Kismet

Test Case No.	Description	Status	Remarks	Correction
1.	Web UI logs and displays alerts in real-time	Failure	Alerts do not appear in /alerts/ directory and dashboard interface	Check Kismet log path.
2.	Kismet alerts persist in database/logs	Failure	Alert entries are not saved in .kismet DB or .log or .txt files	Ensure logging is enabled in config
3.	Capture Channel remains fixed	Success	Kismet stays on set channel and doesn't miss packets	-

Table: 5.3.2 Integration Testing with Kismet

Table: Simulation Verification Testing:

Test Case No.	Description	Status	Remarks	Correction
1.	Simulate Deauth Flood using MDK4	Success	Devices disconnect, alerts triggered.	-
2.	Simulate Evil Twin with same SSID	Success	Rogue AP visible in scans, alert fired.	-
3.	Simulate Fake AP Flood with format strings.	Success	Dozens of fake SSIDs flood the spectrum, alert shown in Kismet.	-

Table: 5.3.3 Simulation Verification Testing

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

Wireless intrusion detection and response system (Widrs) is designed as a solid and cheap solution for wireless network protection against various cyber attacks. As many rely on wireless connection in domestic surroundings, public places and office sites, cost -effective security system with real -time defense options requires pressure.

Widrs runs on light platforms such as Raspberry Pie and Ubuntu-based platforms to enable heavy distribution in different environments without the need for expensive infrastructure. Depending on the surveillance of real -time and smart detection, the system detects attacks such as hiring attacks, evil twin configurations, ARP falsification and denial of service attacks. By using devices such as Aircrack-Ng, Snort, TCPDump and Iptables, Widr's high recognition accuracy provides attacks.

The modular design allows real -time reaction functions such as blocking the attacker's IP, instant notification and separation of malicious users from the network. With automated control and user -friendly interfaces, administrators can lock the wireless environment with minimal intervention.

In essence, wireless network security increases through a comprehensive solution to prevention, detection and reaction, so reducing the weaknesses and improves the general cyber security currency.

6.2 Future Scope:

In the not-so-distant future, WIDRS has a very good future. The more wireless networking is a part of our way of life and workplace, the more necessary it is to maintain an intelligent, dynamic security solution like WIDRS. Perhaps the most fascinating possible potential is to have the system become more intelligent over a period of time. The more information pass through it, the more WIDRS knows and becomes better—detecting suspicious activity, new types of attacks, and even previously unseen attacks. It won't simply respond—it'll predict. There is also vast potential to scale WIDRS. Whether it is a corporate or university campus, or even a smart city, WIDRS can be implemented at multiple locations and collaborate to secure everything. Cloud technology and secure logging methodologies will make it efficient and safe.With increasing numbers of gadgets connecting to our networks—televisions, toasters, et al.—WIDRS will be able to identify anything untoward. In the event of a gadget developing rogue behavior or trying to attach itself maliciously, WIDRS can shut it down prior to causing harm.

To make things easier for the security admins, there is also a mobile app being created. This would allow admins to see notifications and reply to problems from their phones, wherever they are.And it doesn't end there. Much of WIDRS's potential for the future lies within the community surrounding it. A place where users and experts can exchange custom rules, threat intelligence, and working solutions in the real world. With the right collaborations—with cybersecurity professionals, researchers, and even other tech developers—WIDRS can just continue to get more powerful.All in all, WIDRS is not only a tool—it's becoming an intelligent, community-driven keeper of the wireless world. And as networks expand and evolve further, WIDRS will be at the forefront, one step ahead of them.

CHAPTER 7

REFERNCES

[1] "Enhanced Security: A Wireless Monitoring System with Advanced Security Features, 2023"

https://www.researchgate.net/publication/387075682_Enhanced_Security_A_Wireless_Monitoring_System_with_Advanced_Security_Features

[2] Performance of Kismet Wireless Intrusion Detection System on Raspberry Pi

https://www.researchgate.net/publication/359605171_Performance_of_Kismet_Wireless_Intrusion_Detection_System_on_Raspberry_Pi

[3] "A Wireless Intrusion Detection System for 802.11 WPA3 Networks"

https://arxiv.org/abs/2110.04259

[4] "Design of a Snort-based IDS on the Raspberry Pi 3 Model, 2021"

https://ieeexplore.ieee.org/document/9527511

[5] "IDS Using Raspberry Pi and Telegram Integration, 2021"

https://dl.acm.org/doi/10.1145/3487923.3487928?utm_source=chatgpt.com

[6] "Indoor Intrusion Detection and Filtering System Using Raspberry Pi, 2020"

https://www.researchgate.net/publication/340690794_Indoor_Intrusion_Detection_and_Filtering_System_Using_Raspberry_Pi

[7] "Raspberry Pi-Based Intrusion Detection System, 2018"

https://www.researchgate.net/publication/387075682_Enhanced_Security_A_Wireless_Monitoring_System_with_Advanced_Security_Features

[8] "Raspberry Pi-Based Investigating Model for Intrusion Evidence, 2018"

https://www.researchgate.net/publication/325408844_Raspberry_Pi-Based_Investigating_Model_for_Identifying_Intrusion_Evidence

[9] "RaspyAir: Self-Monitoring System for Wireless Intrusion Detection using Raspberry Pi, 2017"

https://www.researchgate.net/publication/313844564_RaspyAir_Self-Monitoring_System_for_Wireless_Intrusion_Detection_using_Raspberry_Pi

[10] "Smart WIDS Implementation for SOHO Environment, 2016"

https://www.researchgate.net/publication/310785152_Smart_Wireless_Intrusion_Detection_System_Implementation_for_SOHO_Environment

CSE-A6 GRIET

A6.pdf

Gokaraju Rangaraju Institute of Engineering and Technology

Document Details

Submission ID
trn:oid:::3618:94395068

Submission Date
May 5, 2025, 3:27 PM GMT+5:30

Download Date
May 5, 2025, 3:30 PM GMT+5:30

File Name
A6.pdf

File Size
3.2 MB

42 Pages

6,327 Words

36,653 Characters

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Small Matches (less than 8 words)
- Crossref database
- Crossref posted content database

Match Groups

19 Not Cited or Quoted **3%**
Matches with neither in-text citation nor quotation marks

0 Missing Quotations **0%**
Matches that are still very similar to source material

9 Missing Citation **1%**
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted **0%**
Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
1% Publications
2% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 19 Not Cited or Quoted 3%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 9 Missing Citation 1%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
1% Publications
2% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
eudl.eu		<1%
2	Internet	
krex.k-state.edu		<1%
3	Submitted works	
Midlands State University on 2024-05-22		<1%
4	Internet	
codelivly.com		<1%
5	Internet	
www.e-spincorp.com		<1%
6	Internet	
export.arxiv.org		<1%
7	Internet	
elibrary.stipram.ac.id		<1%
8	Internet	
lbms03.cityu.edu.hk		<1%
9	Internet	
www.iieta.org		<1%
10	Internet	
www.mdpi.com		<1%

11 Internet

www.wevolver.com <1%

12 Publication

Madzin, Hizmawati. "Computer-Generated Fractals with Medical Application in Tr... <1%

13 Internet

certsimple.com <1%

14 Internet

dspace.dtu.ac.in:8080 <1%

15 Submitted works

The Hong Kong Polytechnic University on 2006-04-09 <1%

16 Internet

hal.archives-ouvertes.fr <1%

17 Submitted works

Asia Pacific University College of Technology and Innovation (UCTI) on 2023-09-23 <1%

18 Submitted works

University of Sydney on 2020-03-28 <1%

19 Internet

dailygram.com <1%

20 Internet

www.coursehero.com <1%

Wirelsess Intrusion Detection

Dr. P. Vara Prasada Rao
Dept. of CSE, GRIET
Hyderabad, India
ykrishna915@grietcollege.com

G. Sai Aditya
Dept. of CSE, GRIET
Hyderabad, India
aditya22241a0532@grietcollege.com

K. Naveen
Dept. of CSE, GRIET
Hyderabad, India
naveen22241a0540@grietcollege.com

B. Iok Vardhan
Dept. of CSE, GRIET
Hyderabad, India
vardhan22241a0514@grietcollege.com

K. Sai Charan
Dept. of CSE, GRIET
Hyderabad, India
charan22241a0537@grietcollege.com

B. Vamshi
Dept. of CSE, GRIET
Hyderabad, India
vamshi22241a0513@grietcollege.com

Abstract—Wireless networks are highly vulnerable to cy-berattacks that can compromise network integrity, intercept confidential data, and disrupt connectivity. Common threats such as Deauthentication Attacks, Evil Twin Attacks, ARP Spoofing, and Wi-Fi-based DDoS Attacks pose significant risks to both individual and enterprise networks. These attacks often lead to unauthorized access, data breaches, and denial-of-service inci-dents, necessitating proactive cybersecurity measures. To mitigate these risks, this project proposes the implementation of a Wireless Intrusion Detection and Response System (WIDRS) on Raspberry Pi or Ubuntu-based operating systems. The system continuously monitors wireless traffic using advanced tools like Aircrack-ng, iptables, Fail2Ban, Snort, and tcpdump. Upon detecting anoma-lous activity, it initiates automated responses such as sending alerts via email and Telegram, blocking malicious IP addresses, and deauthenticating rogue users to prevent further exploitation. This approach delivers strong performance in various environ-ments, including public Wi-Fi, corporate networks, home setups, and IoT ecosystems. By enabling real-time threat detection and automated mitigation, WIDRS enhances network security and ensures a resilient wireless communication infrastructure.

Index Terms—Wireless Security,Intrusion Detection System (IDS), Cybersecurity, Automated Threat Mitigation, Raspberry Pi.

I. Introduction

In a time of fast-paced technological advancements and ubiquitous connectivity, wireless networks have emerged as a cornerstone of modern communications networks. Wireless networks make easy information sharing possible, support real-time collaboration, and provide mobility in ways unseen before. From homes and homes for learning to businesses and government, wireless access is no longer a luxury but a neces-sity. But growing wireless technology usage has also opened doors to a vast array of security vulnerabilities and loopholes. Unlike wire-based networks where access is physically lim-ited, wireless networks employ open airwaves and thus are in-herently more vulnerable to cyberattacks.The broadcast nature of wireless communication renders the information susceptible to being easily eavesdropped on, compromised, or injected with unauthorized data packets without any physical hardware access within the network needed. Such ease of exploitation has given birth to a sharp increase in wireless-specific threats like eavesdropping, rogue access points, session hijacking, denial-of-service (DoS) attacks, and unauthorized outsider access. Furthermore, as wireless networks become increasingly sophisticated—frequently integrating Internet of Things (IoT) devices, mobile devices, and cloud-connected services—the attack surface increases exponentially. Standard security con-trols like firewalls and antivirus products, though still needed, are usually not enough to detect or respond to sophisticated and evasive attacks in the wireless realm. This immediately calls for advanced systems that not only alert on intrusions but also respond to them in real time.

AI. Literature Review

[1] discusses the design and theoretical implementation of a secure wireless monitoring system using ESP8266 modules with encryption protocols like AES and SHA-based authen-tication. Real-time alerts and encrypted data transmission to a central node ensure confidentiality, though the system lacks practical large-scale deployment and power optimiza-tion. [2] explores deploying Kismet WIDS on Raspberry Pi to provide low-cost intrusion detection for household and SOHO networks. The system performs well in real-time traffic monitoring, but it lacks advanced techniques like behavioral analysis or real-time auto-response. [3] develops a WIDS for 802.11 WPA3 networks using a custom packet analyzer target-ing handshake anomalies and DoS attack patterns. Although it leverages WPA3-specific features effectively, it remains untested in mixed or real-world scenarios. [4] demonstrates Snort-based IDS on Raspberry Pi 3 for detecting port scans and brute-force attacks in small networks. It performs well under light load but suffers from performance drops under high traffic and lacks adaptive detection features. [5] presents an IDS using Raspberry Pi integrated with Telegram for real-time alerts. Python-powered and lightweight, it suits small setups, although its dependency on constant connectivity makes it unsuitable for delay-sensitive environments. [6] introduces an indoor IDS using Raspberry Pi to detect malicious QR code scans via Python-based filters. While effective indoors, it is not suited for broader wireless intrusion detection tasks.

[7] analyzes a Raspberry Pi-based IDS using Wireshark and

Ettercap to detect ARP spoofing and MAC flooding. It's low-cost and viable for home networks, but lacks scalability and cannot handle advanced threats without future enhancements like machine learning. [8] proposes a forensic evidence col-lection model using Raspberry Pi to log intrusion data for post-incident analysis. Though cost-effective for forensics, it does not support real-time detection or prevention. [9] introduces RaspyAir, a Raspberry Pi-based wireless IDS that flags abnormal signal patterns. Suitable for basic detection, it struggles with scalability and does not protect against DoS or complex attacks. [10] outlines a Snort-based WIDS for SOHO environments using Raspberry Pi 2 with web interface inte-gration. While it offers timely alerts and affordability, its high false-positive rate and lack of centralized threat management limit its application to larger networks.

Gaps Identified :

- Lack of large-scale real-world deployment and evaluation in dynamic environments.
- Minimal or no integration of advanced detection tech-niques such as machine learning or behavioral analysis.
- Limited or no real-time automated response capabili-ties—mostly reactive systems.
- High dependency on signature-based detection, making systems ineffective against zero-day or unknown threats.
- Inadequate power management strategies for long-term operation on low-power devices like Raspberry Pi.
- Absence of scalability in terms of handling high network traffic or multiple access points.
- Lack of centralized management for handling multiple devices or broader networks.
- Minimal consideration for mixed protocol environments (e.g., WPA2 and WPA3 coexistence).
- Poor support for complex attack types such as DDoS or advanced persistent threats.

BI. Proposed System

Current advancements in wireless technology have high-lighted the vulnerability of wireless networks to cy-berattacks due to their open accessibility. Community wireless networks, in particular, are often targeted due to insufficient security measures. To address these chal-lenges, the Wireless Intrusion Detection and Response System (WIDRS) offers a real-time, proactive security mechanism capable of monitoring and mitigating a wide range of wireless threats. Tested on lightweight plat-forms such as Raspberry Pi and Ubuntu-based systems, WIDRS has demonstrated effectiveness in detecting and responding to attacks such as Deauthentication Floods, ARP Spoofing, Evil Twin exploits, and Denial-of-Service (DoS) incidents.

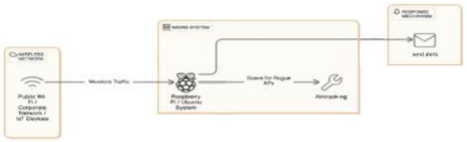


Fig. 1. WIDRS System Architecture

The system utilizes standardized tools including Aircrack-ng, tcpdump, Snort, and iptables to continuously monitor wireless traffic and collect relevant data. This data is then analyzed to detect abnormal activities, upon which the system triggers automated responses such as blocking the attacker's IP address, alerting users, and disconnecting unauthorized clients. WIDRS is built using a modular architecture comprising distinct components for monitoring, detection, response, and logging. This design enhances the system's manageability, allows for seamless updates, and ensures scalability across various deployment environments.

IV. Implementation

The Wireless Intrusion Detection System (WIDS) was implemented on a Raspberry Pi to provide a lightweight, low-cost, and efficient platform for monitoring and responding to wireless threats in real time. The imple-mentation was divided into two parts: the monitoring setup and the attacking simulation environment to evaluate the system's capabilities.

A. System Setup

The Raspberry Pi was configured with Kali Linux as the base operating system. The system was updated and upgraded to ensure compatibility with the latest wireless security tools. A USB Wi-Fi adapter supporting monitor mode (Atheros AR9271 chipset) was connected for packet capture.

Key tools installed and configured include:

- Kismet: For wireless traffic capture and intrusion detec-tion.
- Airmon-ng: To enable monitor mode on the wireless adapter.
- Tcpdump: For low-level packet capture.
- MDK4: To simulate deauthentication and rogue access point attacks.

B. Monitoring Configuration

The adapter was switched to monitor mode using airmon-ng, and conflicting processes were killed using airmon-ng check kill. iwconfig was used to verify the adapter status. Kismet was launched to start real-time

monitoring of all wireless traffic, and its dashboard was used to visualize detected devices and generate alerts.

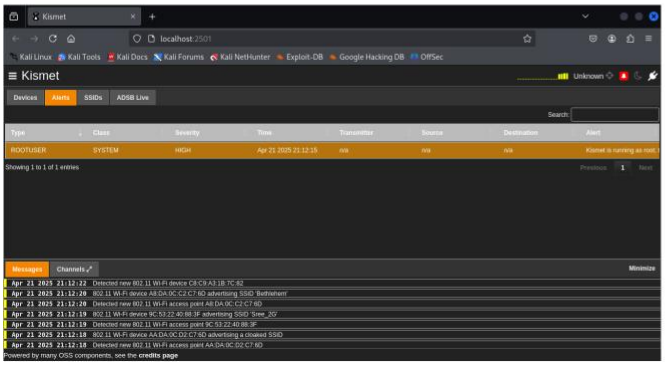


Fig. 2. Kismet Dashboard displaying captured wireless activity

C. Attack Simulation

A separate device running Kali Linux simulated various attacks:

- Deauthentication Floods: Using MDK4 to disconnect devices from access points.
- Evil Twin Attack: By broadcasting fake access points with duplicate SSIDs.
- Fake Access Point Flood: Dozens of bogus SSIDs were injected into the airspace.

These attacks were used to evaluate the effectiveness of the WIDRS in recognizing and responding to threats.

D. Detection

Kismet's alert system successfully detected multiple forms of wireless attacks:

- DEAUTH/FLOOD
- SPOOF (Evil Twin indicators)
- FORMATSTRING (Fake AP injection)

V. Results

To validate the effectiveness of the Wireless Intrusion Detection and Response System (WIDRS), various wireless attacks were simulated using a dedicated attacking device. The system's detection and alert mechanisms were monitored through Kismet's interface. Below are the documented results of each test scenario.

A. Deauthentication Flood Detection

A deauthentication flood was launched using MDK4 to forcibly disconnect devices from the access point. The WIDRS system successfully identified the threat and raised alerts clas-sified under DEAUTH/FLOOD, indicating a denial-of-service attempt.

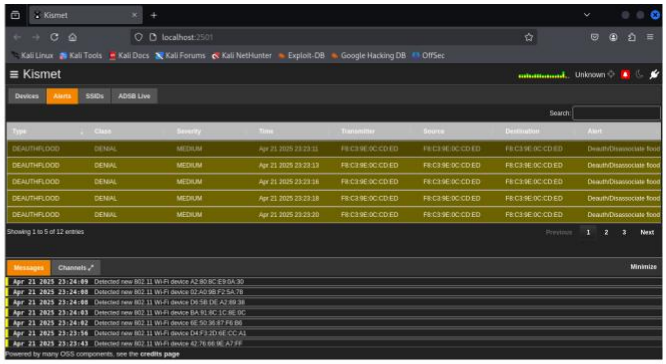


Fig. 3. Alert for Deauthentication Flood attack

B. Fake Access Point Flood Detection

A fake AP flood was simulated by broadcasting numerous bogus SSIDs to saturate the wireless spectrum. The system detected this behavior as an exploit and raised high-severity alerts labeled FORMATSTRING and ROOTUSER.

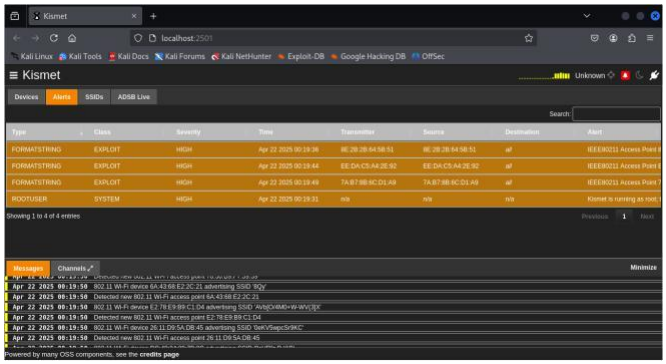


Fig. 4. Detection of Fake Access Point Flood using Kismet

C. Evil Twin Attack Detection

An Evil Twin attack was performed by cloning a legitimate access point's SSID and broadcasting it using a separate interface. Kismet detected spoofing behavior, marking it with SPOOF class alerts such as NOCLIENTMP.

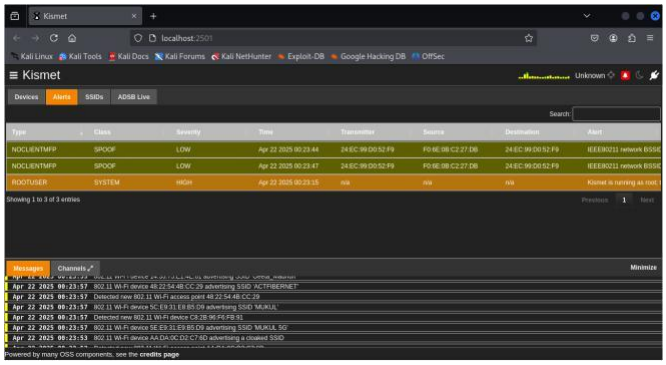


Fig. 5. Evil Twin attack alert displayed in Kismet

VI. Conclusion and Future Scope

Wireless Intrusion Detection and Response System (WIDRS) is an affordable and effective solution to protect wireless networks from various cyberattacks. With the growing dependence on wireless connectivity in homes, offices, and public spaces, WIDRS addresses the need for cost-effective security systems that provide real-time protection. The system runs efficiently on lightweight platforms such as Raspberry Pi and Ubuntu-based systems, allowing widespread deployment without requiring high-end infrastructure. By continuously monitoring the network, WIDRS can detect threats like deauthentication floods, evil twin setups, ARP spoofing, and denial-of-service attacks. Using tools like Aircrack-ng, Snort, tcpdump, and iptables, the system offers high accuracy in identifying malicious activity. It also features real-time response actions such as IP blocking, user alerts, and unauthorized user disconnections, with a user-friendly interface for easy management.

Looking to the future, WIDRS is positioned to evolve as wireless networking becomes more integral in daily life. As the need for intelligent and adaptive security solutions increases, WIDRS will be able to evolve into a smarter system capable of learning from traffic patterns and adapting to emerging threats. This would allow it to not only respond to known attacks but also predict and prevent new or unknown intrusions. The system's scalability is another strength, as it can be deployed across diverse environments such as corporate offices, universities, and smart cities. Cloud integration and secure logging will enhance performance and data integrity. With the rise of IoT devices, WIDRS will be able to detect and isolate rogue devices, preventing potential threats before damage occurs. A mobile app is also being developed to allow administrators to monitor and respond to incidents remotely.In the long term, WIDRS has the potential to transform into a community-driven platform, where users, experts, and devel-opers collaborate to exchange threat intelligence and innova-tive solutions. As it continues to evolve, WIDRS will shift from being a simple security tool to an intelligent, adaptive guardian of wireless networks. With ongoing development and strategic partnerships, it will remain a cutting-edge solution, continuously staying ahead of emerging cybersecurity threats in an increasingly connected digital world.

References

[1] M. Sufyan, M. Jameel, et al., "Enhanced Security: A Wireless Monitoring System with Advanced Security Features," 2023.

[2] A. Al-Attar and O. Aldabbas, "Performance of Kismet Wireless Intrusion Detection System on Raspberry Pi," 2022.

[3] N. Dalal, N. Akhtar, and A. Gupta, "A Wireless Intrusion Detection System for 802.11 WPA3 Networks," 2021.

[4] L. D. C. Silveira, L. A. G. Silva, and D. S. Freitas, "Design of a Snort-based IDS on the Raspberry Pi 3 Model," 2021.

[5] H. Wijaya and E. G. Rachma, "IDS Using Raspberry Pi and Telegram Integration," 2021.

[6] R. Deshmukh, "Indoor Intrusion Detection and Filtering System Using Raspberry Pi," 2020.

[7] B. G. Rasane and S. S. Mane, "Raspberry Pi-Based Intrusion Detection System," 2018.

[8] A. K. Sharma and K. B. Anoop, "Raspberry Pi-Based Model for Investigating Intrusion Evidence," 2018.

[9] J. V. Divya and N. Niranjana, "RaspyAir: Self-Monitoring System for Wireless Intrusion Detection using Raspberry Pi," 2017.

[10] A. R. Andriansyah and A. Setiawan, "Smart WIDS Implementation for SOHO Environment," 2016.