iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.83072

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

An Embedded Wireless Intrusion Detection System Architecture for Securing WPA2-Connected Medical Devices

K. Vani[1], M. Thanushhri[2]
[1]*Assistant Professor, Department of Computer Applications, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India*
[2]*Student, Department of Computer Science and Engineering (Cyber Security), VIT Bhopal University, Madhya Pradesh, India*

Abstract: ***Wireless networks remain a cornerstone of modern embedded, medical, industrial, and Internet of Things (IoT) systems, valued for their flexibility, scalability, and ease of deployment. Although WPA3 has emerged as the latest security standard, Wi-Fi Protected Access II (WPA2) continues to dominate in legacy and resource-constrained devices due to hardware compatibility constraints and migration costs. However, WPA2 is vulnerable to multiple security threats, including deauthentication attacks, management frame spoofing, weak authentication schemes, and exploits such as the Key Reinstallation Attack (KRACK). These vulnerabilities undermine the reliability and confidentiality of wireless communications, particularly in embedded systems supporting critical applications.***

This research introduces a practical framework to enhance WPA2 security in embedded wireless systems without requiring hardware replacement. The approach integrates WPA3-inspired protections—such as IEEE 802.11w Protected Management Frames (PMF), enforced AES-CCMP encryption, secure firmware and driver configurations, and strengthened authentication practices. Implementation is carried out on an embedded Linux platform using the ATWILC3000 Wi-Fi module with an i.MX6ULL processor running Linux Kernel 4.1.15. Configuration and validation employ hostapd, wpa_supplicant, cfg80211, and Wireshark-based packet analysis. Experimental results show that enabling PMF significantly improves resistance to spoofed deauthentication and disassociation attacks while preserving WPA2 compatibility. Enhanced encryption policies and secure configuration practices further bolster network resilience. These findings demonstrate that substantial security gains can be achieved in legacy WPA2 deployments through software-level modifications and protocol hardening, thereby extending the secure operational lifespan of embedded wireless systems. This work offers a cost-effective, practical methodology for strengthening WPA2 in embedded environments where full migration to WPA3 is not yet feasible.

Keywords: Wireless networks, Wi-Fi Protected Access, deauthentication, spoofing, Key Reinstallation Attack, embedded wireless systems, Protected Management Frames, encryption.

I. INTRODUCTION

Wireless communication technologies have become an essential part of modern digital infrastructure due to their flexibility, mobility, and ease of deployment. Wi-Fi is the most frequently used means of communicating data wirelessly in a fixed location.[1] IEEE 802.11 standard defines communication among networks at the MAC layer by exchanging three frames: viz, control frames, data frames, and management frames. [2] Among the various wireless security protocols developed for protecting Wi-Fi communications, Wi-Fi Protected Access II (WPA2) has remained one of the most extensively deployed standards for securing wireless local area networks (WLANs). WPA2 provides confidentiality, integrity, and authentication through the implementation of the IEEE 802.11i standard and the use of Advanced Encryption Standard Counter Mode Cipher Block Chaining Message Authentication Code Protocol (AES-CCMP).

WPA 2 supports two modes of security viz “home User” and “Corporate User”. A pre-shared passphrase or passkey is used in home user mode, and Access points are manually configured for the authentication [3]. Despite its widespread adoption, WPA2 suffers from several security limitations and vulnerabilities that can compromise wireless communications. Over the years, researchers and security analysts have demonstrated multiple attacks against WPA2 networks, including deauthentication attacks, disassociation attacks, dictionary-based password cracking, Evil Twin attacks, PMKID attacks, and the Key Reinstallation Attack (KRACK). One of the major weaknesses of traditional WPA2 implementations is the lack of mandatory protection for management frames, allowing attackers to spoof deauthentication or disassociation frames and force wireless clients to disconnect from legitimate networks.

Such attacks can significantly affect the reliability and availability of wireless communication systems, especially in embedded and mission-critical environments such as medical devices and industrial control systems.

To address the growing security concerns in wireless networks, the Wi-Fi Alliance introduced WPA3 which addresses the inherent vulnerabilities of previous protocols by implementing more robust cryptographic algorithms and enhanced authentication protocols, thereby mitigating risks such as dictionary attacks and ensuring forward secrecy. [4] However, the migration from WPA2 to WPA3 is challenging for many embedded and legacy systems due to hardware limitations, firmware compatibility issues, computational constraints, and increased deployment costs. As a result, a large number of embedded wireless devices continue to operate using WPA2-based infrastructure.

In many real-world applications, especially in embedded Linux systems, replacing wireless hardware to support WPA3 is not economically or technically feasible. Therefore, improving the security of existing WPA2 deployments through software-level enhancements and protocol hardening techniques has become an important area of research. This research focuses on enhancing WPA2 security in embedded wireless systems by incorporating selected WPA3-inspired security mechanisms without requiring major hardware modifications. The proposed approach includes the implementation of Protected Management Frames (PMF) based on IEEE 802.11w, enforcement of AES-CCMP encryption, secure firmware and driver configuration, and strengthened wireless authentication practices.

The implementation and analysis presented in this research are performed using the ATWILC3000 Wi-Fi module integrated with the i.MX6ULL processor running Linux Kernel 4.1.15. The Linux wireless stack components, including cfg80211, hostapd, and wpa_supplicant, are configured and analysed to study the feasibility of implementing enhanced security mechanisms in resource-constrained embedded systems. Wireshark-based packet analysis is used to validate the behaviour of management frame protection and to observe the differences between standard WPA2 communication and enhanced WPA2 security configurations.

The primary objective of this research is to develop a practical and cost-effective framework for improving WPA2 security in embedded wireless environments while maintaining compatibility with existing hardware platforms. By implementing PMF and strengthening wireless security configurations, this work aims to reduce the susceptibility of WPA2 networks to spoofing and management frame attacks. The study also evaluates the effectiveness of these enhancements in improving the resilience, reliability, and overall security posture of embedded wireless systems.

II. BACKGROUND AND LITERATURE REVIEW

Wireless communication technologies based on the IEEE 802.11 standards have become the foundation of modern wireless networking systems. As wireless communication became more common, securing wireless transmissions against unauthorized access and attacks became a major research concern. Wireless network security is conventionally achieved through cryptographic protocols at multiple layers in the network stack, including IPSecurity, Wi-Fi Protected Access, and Secure Sockets Layer [5]. The development of wireless security protocols evolved from Wired Equivalent Privacy (WEP) to Wi-Fi Protected Access (WPA) and eventually to Wi-Fi Protected Access II (WPA2), which became the industry standard for wireless security after the ratification of IEEE 802.11i in 2004. WPA2 introduced major improvements over WEP and WPA by implementing stronger authentication and encryption mechanisms. The protocol primarily relies on the IEEE 802.11i security framework and uses AES-CCMP encryption to ensure confidentiality, integrity, and authentication of wireless traffic. WPA2 operates in two modes: WPA2-Personal, which uses a Pre-Shared Key (PSK), and WPA2-Enterprise, which uses IEEE 802.1X authentication with authentication servers. Although WPA2 significantly improved wireless security compared to earlier standards, several vulnerabilities and implementation weaknesses were discovered over time. One of the most significant WPA2 vulnerabilities is the Key Reinstallation Attack (KRACK), discovered by Mathy Vanhoef in 2017. KRACK exploits weaknesses in the WPA2 four-way handshake procedure by manipulating retransmitted handshake messages, causing cryptographic key reinstallation and nonce reuse. This allows attackers to replay, decrypt, and in some cases forge packets within wireless communications. The vulnerability affected millions of Wi-Fi devices, particularly embedded and IoT systems that lacked timely firmware and software updates. Apart from KRACK, wireless networks remain vulnerable to several management frame attacks such as deauthentication and disassociation attacks. Traditional WPA2 implementations do not protect management frames, allowing attackers to spoof deauthentication packets and force legitimate users to disconnect from wireless networks. These attacks are widely used in denial-of-service scenarios and in advanced attacks such as Evil Twin attacks and Multi-Channel Man-in-the-Middle attacks. Recent studies have shown that such attacks continue to affect both WPA2 and WPA3 environments under certain conditions. The presence of rogue access points is also a vulnerability. It broadcasts the legitimate SSID, allowing the legitimate device to connect to the fake access point with a legitimate pre-Shared Key.[2] Once the key is obtained, anyone could enter the network.

To address management frame vulnerabilities, IEEE introduced the IEEE 802.11w amendment, commonly known as Protected Management Frames (PMF) or Management Frame Protection (MFP). PMF provides integrity and authentication protection for selected management frames, thereby reducing the effectiveness of spoofed deauthentication and disassociation attacks. Research studies have shown that PMF significantly improves wireless network resilience, particularly in embedded and IoT environments where denial-of-service attacks can critically affect system reliability.

However, several studies also indicate that PMF alone is not sufficient to eliminate all wireless security threats. Researchers have identified weaknesses in management frame handling and demonstrated attacks capable of bypassing certain PMF protections under specific implementation conditions. Studies on WPA2 and WPA3 management frame vulnerabilities have revealed that improper implementation of PMF can still expose wireless systems to denial-of-service attacks.

Another major concern in WPA2 deployments is the continued use of legacy encryption protocols such as TKIP. Although WPA2 supports AES-CCMP, many legacy systems still maintain backward compatibility with TKIP for older devices. Security researchers and industry organizations strongly discourage the use of TKIP because of its known cryptographic weaknesses and susceptibility to replay and packet injection attacks. Modern wireless security recommendations encourage exclusive use of AES-CCMP in WPA2 deployments.

Embedded systems and IoT devices face unique security challenges that set them apart from traditional computing platforms. Many of these devices run on older operating system kernels and outdated firmware, and they often have limited processing power and memory. Because of these constraints, they can go for long periods without receiving critical security updates. This makes them appealing targets for attackers, especially those using wireless exploits. Research into IoT wireless security shows that delays in firmware updates and incomplete adoption of Protected Management Frames (PMF) leave many devices vulnerable to well-known threats such as KRACK, FragAttacks, and multi-channel man-in-the-middle attacks.

Several recent research efforts have focused on improving wireless security through intrusion detection systems, machine learning techniques, and enhanced wireless monitoring. Wireless intrusion detection systems have been proposed to detect attacks such as deauthentication flooding, beacon flooding, and rogue access point behavior in WPA2 and WPA3 networks. Zhang *et al*. [6] proposed using the MAC filtering mechanism where a smart client can differentiate between legitimate and non-legitimate frames. Machine learning approaches are also being explored for adaptive wireless security monitoring and anomaly detection in IEEE 802.11 networks.

While WPA3 offers stronger protections—such as Simultaneous Authentication of Equals (SAE), mandatory PMF, and forward secrecy—many embedded systems struggle to adopt it fully due to hardware and firmware limitations. As a result, improving the security of existing WPA2-based systems has become a practical and cost-effective focus for many organizations. Current best practices include enabling PMF wherever possible, enforcing AES-CCMP encryption, patching known vulnerabilities in wireless supplicant software, and ensuring firmware integrity. Together, these measures can significantly improve the security posture of legacy devices without requiring expensive hardware replacements.

The present research builds upon these findings by focusing on the implementation of WPA3-inspired security mechanisms within WPA2-based embedded Linux systems. The proposed framework specifically targets embedded hardware platforms using the ATWILC3000 Wi-Fi module integrated with the i.MX6ULL processor running Linux Kernel 4.1.15. By implementing PMF, secure wireless configuration practices, and enhanced driver and firmware security mechanisms, this work aims to provide a practical and cost-effective approach for improving WPA2 security without requiring complete hardware replacement.

III. VULNERABILITIES AND SECURITY CHALLENGES IN WPA2 NETWORKS

Wi-Fi Protected Access II (WPA2) was introduced as a major improvement over previous wireless security mechanisms such as Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA). WPA2 enhanced wireless security through the use of IEEE 802.11i standards, AES-CCMP encryption, and improved authentication mechanisms. Despite these advancements, WPA2 networks remain vulnerable to multiple security threats and implementation weaknesses that continue to affect embedded systems, enterprise networks, IoT devices, and wireless communication infrastructures. Over time, researchers and cybersecurity analysts have identified several attack vectors capable of compromising the confidentiality, integrity, and availability of WPA2-protected networks.

One of the most common vulnerabilities in WPA2 networks is the deauthentication attack. In traditional IEEE 802.11 networks, management frames such as deauthentication and disassociation frames are transmitted without cryptographic protection. As a result, attackers can spoof these management frames and force legitimate wireless clients to disconnect from an access point.

Deauthentication attacks are widely used in denial-of-service attacks and are frequently combined with credential-harvesting attacks such as Evil Twin attacks.[7] Since WPA2 does not mandate protection of management frames, these attacks remain highly effective in networks where Protected Management Frames (PMF) are not enabled.

Another major WPA2 vulnerability is the Key Reinstallation Attack (KRACK), discovered by Mathy Vanhoef in 2017. KRACK exploits weaknesses in the WPA2 four-way handshake process by forcing victims to reinstall previously used encryption keys. During retransmission of handshake messages, attackers manipulate nonce values and replay packets, resulting in nonce reuse and weakened encryption security. For a KRACK attack to succeed, the hacker needs to be close to the target. The proximity is necessary because the target and the hacker have to share the same Wi-Fi network. [8] This vulnerability allows attackers to decrypt network traffic, replay packets, and potentially inject malicious data into wireless communications.

WPA2 networks are also vulnerable to dictionary and brute-force attacks due to weak Pre-Shared Keys (PSKs). In WPA2-Personal mode, network security heavily depends on the strength of user-defined passwords. Attackers can capture the WPA2 four-way handshake and perform offline dictionary attacks to recover weak passwords. Since the handshake process itself is not encrypted, attackers can repeatedly attempt password combinations without directly interacting with the access point. Weak passwords and poor credential management practices therefore remain one of the primary causes of WPA2 compromise.[9]

PMKID attacks represent another significant weakness in WPA2 networks. In these attacks, attackers capture the Pairwise Master Key Identifier (PMKID) directly from the access point without requiring a connected client. The captured PMKID can then be used for offline password cracking attacks. PMKID attacks simplify wireless credential attacks because attackers no longer need to wait for a legitimate client to reconnect or perform a deauthentication attack to capture handshake data. [10] This method became increasingly popular after tools such as Hashcat and hcxdumptool introduced automated PMKID capture and cracking capabilities.

Another serious security threat is the Evil Twin attack. In this attack, attackers create a rogue access point that imitates a legitimate wireless network by copying the same SSID, channel configuration, and security settings. Victims unknowingly connect to the rogue access point, allowing attackers to intercept traffic, steal credentials, or launch further attacks.[11] Evil Twin attacks are commonly combined with deauthentication attacks to force users away from legitimate networks. Embedded and IoT systems are particularly vulnerable because many devices automatically reconnect to previously known networks without validating the authenticity of the access point.

Management frame vulnerabilities are further exploited through beacon flooding and association flooding attacks. In beacon flooding attacks, attackers transmit large numbers of fake beacon frames to overwhelm wireless clients and confuse network discovery mechanisms. Association flooding attacks attempt to exhaust access point resources by continuously sending association requests. These attacks reduce network availability and can significantly degrade wireless performance in embedded and industrial environments. Since many low-resource embedded devices lack advanced intrusion detection capabilities, they are especially susceptible to such attacks. [12]

The continued use of legacy encryption mechanisms such as TKIP also contributes to WPA2 insecurity. Although WPA2 supports AES-CCMP encryption, many devices retain TKIP compatibility for backward support with older hardware. TKIP relies on outdated cryptographic mechanisms and is vulnerable to replay attacks, packet injection, and key recovery attacks. Modern wireless security guidelines strongly discourage the use of TKIP and recommend exclusive use of AES-CCMP. WPA3 completely removes TKIP support because of these security weaknesses. Embedded wireless systems face additional challenges because they often operate with outdated Linux kernels, unpatched firmware, and limited processing capabilities. Resource-constrained devices frequently prioritize low power consumption and cost reduction over security implementation. As a result, many embedded systems continue to operate with vulnerable drivers, incomplete PMF support, and outdated supplicant implementations. [13] Furthermore, firmware updates in embedded environments are often infrequent or entirely absent, increasing long-term exposure to wireless attacks.

Another emerging concern is the rise of FragAttacks (Fragmentation and Aggregation Attacks), which exploit weaknesses in Wi-Fi frame fragmentation and aggregation mechanisms. FragAttacks affect multiple Wi-Fi security protocols, including WPA2 and WPA3, and can allow attackers to inject malicious packets into wireless networks under certain conditions. Although firmware patches can mitigate these attacks, many embedded systems remain vulnerable because of delayed firmware update cycles. [14]

Wireless attacks are getting more complicated these days, and it feels like sticking with the usual WPA2 setup just doesn't cut it for keeping modern embedded systems safe. Those vulnerabilities we talked about earlier really point out how we need something extra to build up resilience in wireless connections, but without having to jump straight to WPA3 all at once. I think things like Protected Management Frames or PMF could make a difference, along with making sure AES-CCMP encryption is enforced properly, and then there's strengthening the firmware to avoid tampering. Improving how authentication works on wireless networks seems practical too, especially for setups that are already running WPA2. It might not solve everything right away.

This chapter went over the big issues and ways attacks hit WPA2 networks, mostly in those embedded environments where resources are tight. Some parts get a bit tricky to explain fully. The next chapter is going to cover this framework I came up with, pulling in some ideas from WPA3 protections along with software tweaks to handle those vulnerabilities better.

IV. PROPOSED SECURITY ENHANCEMENT FRAMEWORK FOR WPA2 NETWORKS

The increasing number of wireless attacks targeting WPA2 networks demonstrates the need for improved security mechanisms in existing wireless infrastructures. Although WPA3 introduces advanced protections such as Simultaneous Authentication of Equals (SAE), mandatory Protected Management Frames (PMF), and improved cryptographic procedures, many embedded and legacy systems are unable to migrate fully to WPA3 due to hardware limitations, firmware constraints, compatibility issues, and deployment costs. Consequently, enhancing the security of existing WPA2 deployments through software-level hardening and protocol improvements has become a practical and economically viable solution.

This research proposes the implementation of a lightweight Intrusion Detection System (IDS) combined with wireless packet monitoring to enhance the cybersecurity of WPA2-based embedded medical devices.

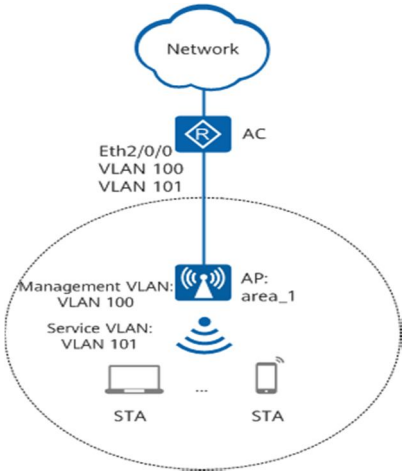
The proposed system continuously monitors wireless traffic, analyses packet behaviour, and detects malicious activities such as deauthentication attacks, rogue access points, packet flooding, and unauthorized connection attempts.

Unlike hardware-dependent security upgrades, the proposed approach operates through software-level monitoring and traffic analysis, making it suitable for legacy medical devices already deployed in healthcare environments.

The experimental platform uses the ATWILC3000-MR110CA Wi-Fi module integrated with the i.MX6ULL processor running Linux Kernel 4.1.15.

A. Existing WPA2 based Medical Device Architecture

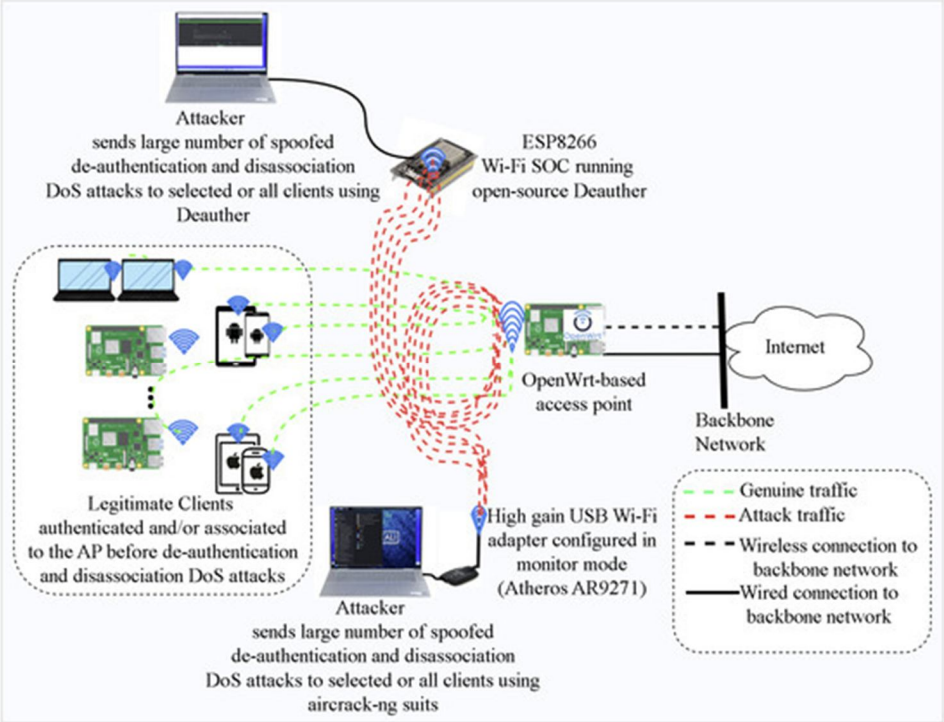
The existing medical device communicates over Wi-Fi using WPA2-AES security. The wireless communication stack consists of: Embedded medical application, WPA supplicant, Linux cfg80211 subsystem, ATWILC3000 driver, Wi-Fi firmware, hardware transceiver. The WPA2 protocol secures data packets using AES-CCMP encryption and authentication through the four-way handshake process.



B. System Architecture

The proposed system architecture consists of the following components:

1) Embedded Medical Device
2) WPA2 Wireless Communication Layer
3) Packet Capture Module
4) Intrusion Detection Engine
5) Traffic Analysis Module
6) Alert and Logging System
7) The IDS passively monitors wireless communication without interrupting normal device operation.



a) Step 1: Wireless Network Initialization

The embedded medical device initializes the wireless communication interface using the ATWILC3000-MR110CA Wi-Fi module connected to the i.MX6ULL processor. During initialization,

Linux wireless drivers are loaded
↓
WPA Supplicant starts the WPA2 authentication process
↓
The device scans for available access points
↓
Secure association with the hospital network is established
↓
AES-CCMP encryption keys are generated through the WPA2 four-way handshake

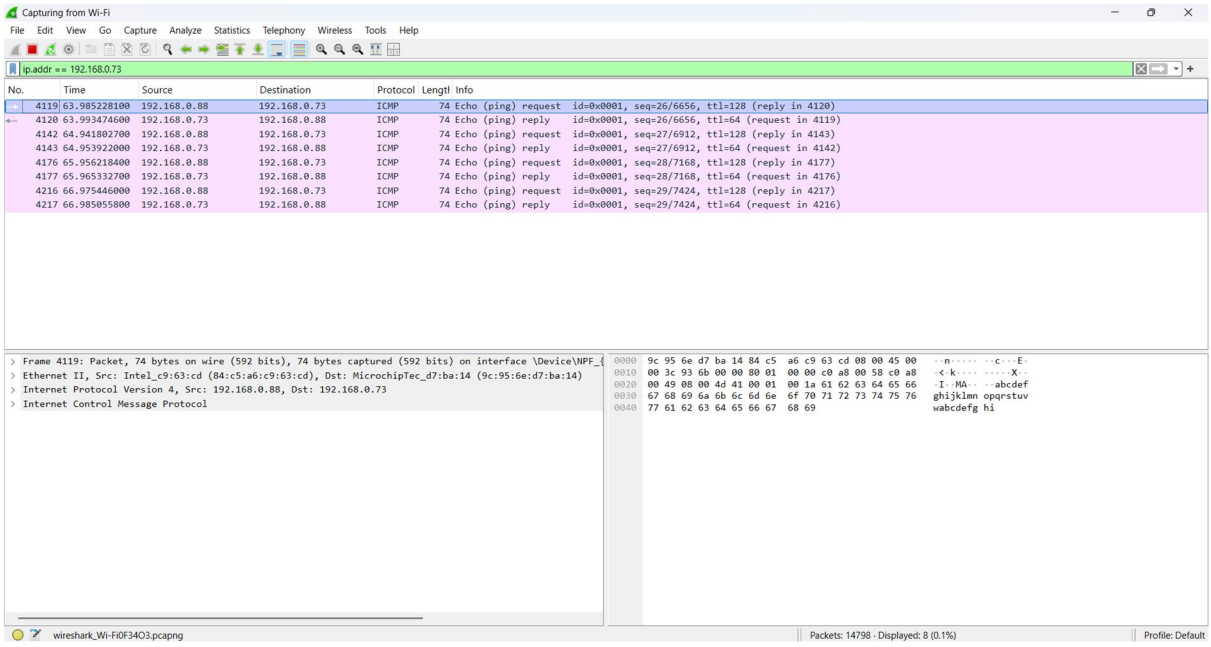
Once the connection is established, the device begins transmitting medical data over the wireless network.

```
C:\Users\rmmoo>ping 192.168.0.73

Pinging 192.168.0.73 with 32 bytes of data:
Reply from 192.168.0.73: bytes=32 time=74ms TTL=64
Reply from 192.168.0.73: bytes=32 time=12ms TTL=64
Reply from 192.168.0.73: bytes=32 time=9ms TTL=64
Reply from 192.168.0.73: bytes=32 time=9ms TTL=64

Ping statistics for 192.168.0.73:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 74ms, Average = 26ms
```

The above image shows the local host pinging the IP address of the medical device.
Once pinged, we can view the incoming IP addresses using any tool like Wireshark.

b) Step 2: Check if the incoming logs have been captured by Wireshark

Wireshark is a powerful network packet analyser that allows users to capture and inspect data packets traveling through a network. It is widely used by network professionals, security engineers and developers due to its versatility and open-source nature.



This image shows how the configuration of both devices' IP addresses was captured.

c) Step 3: Packet Capture Initialization

After network initialization, the IDS activates the packet monitoring subsystem. The packet capture engine performs the following operations:

1. Configures the wireless interface for monitoring
2. Initializes packet capture libraries
3. Allocates packet buffers
4. Starts continuous wireless traffic collection

The IDS passively listens to wireless communication without interrupting normal data transmission.

The following packet categories are monitored:

Beacon frames	Detect nearby access points
Probe Requests	Monitor device discovery behaviour
Authentication frames	Track login attempts
Association frames	Observe connection establishment
Deauthentication frames	Detect forced disconnections
Data frames	Monitor encrypted traffic patterns

d) Step 4: Packet Acquisition and Filtering

Captured packets are transferred to the packet processing module. The filtering engine removes irrelevant traffic and extracts important packet attributes such as Source MAC address, Destination MAC address, Frame type, Signal strength, Timestamp, Sequence Number, SSID information, authentication status.

The filtering process reduces computational overhead by analysing only security relevant packets. Management frames are noticed in focus because they are commonly exploited in WPA2 wireless attacks.

e) Step 5: Traffic Analysis Phase

The filtered packets are analysed by the traffic analysis engine. The analysis module evaluates features like packet transmission frequency, connection behaviour, reassociation patterns, authentication request rates, device communication consistency, wireless signal variations. The system establishes a baseline profile representing normal network behaviour. Any deviation from the baseline is treated as a potential anomaly. The traffic analysis engine continuously compares current traffic patterns against predefined normal behaviour thresholds.

f) Step 6: Intrusion detection engine operation

The detection engine detects network traffic features for potential threats using two approaches:

1. Signature based detection: It iteratively goes through each of the predefined rules, applying the rule's condition to the traffic features. If a rule matches, a signature-based threat is recorded with high confidence.
2. Anomaly based detection: it processes the feature vector (packet size, packet rate and byte rate) through the Isolation Forest Model to calculate an anomaly score. If the score indicates unusual behaviour, the detection engine triggers it as an anomaly and produces a confidence score proportional to the anomaly's severity.

Finally, we return the aggregated list of identified threats with their respective annotation (either signature or anomaly), the rule or score that triggered the anomaly, and a confidence score that suggests how likely it is that the identified pattern is a threat.[15]

g) Step 7: Attack classification

After detecting any suspicious activity, the IDS classifies the attack based on severity and attack type. Attack classification parameters include:

- Packet frequency
- Attack duration
- Network impact
- Device communication interruption
- Repeated attack occurrence

The IDS categorizes threats into Low, Medium, High and Critical based on the monitored activity. Threat prioritization helps administrators respond effectively.

h) Step 8: Alert generation and Logging

Once an attack is confirmed, the IDS generates alerts and stores attack information. The logging system records timestamp, source MAC address, target device, attack type, packet statistics, signal strength, detection status. Alerts may be generated through network monitoring dashboards or log files.

V. IMPLEMENTATION

Step 1: Connecting the devices to the network

We have my laptop which has Wireshark and the IDS installed in it. My laptop is connected to the Wi-Fi here at the medical device manufacturing unit. The medical device is also connected to the same Wi-Fi. Now I ping the IP address of the medical device to my laptop to establish a connection between both devices.

Step 2: Initialize Packet Capture in Wireshark

Once the medical device has been pinged, I open Wireshark and select the network that my device is connected to and start the packet capture. I can see all the data packets that enter the network.

Step 3: Filtering on Wireshark

On the Filter bar, I enter ip. Addr == "IP ADDRESS" and Wireshark filters out all the packets from that particular IP address. Once the packets are shown on the screen save the capture as a pcap file.

Save it as capture.pcap

Step 3: IDS reads the PCAP file

The IDS can analyse packets from the saved capture using Python + Scapy

```
from scapy.all import rdpcap
packets = rdpcap("capture.pcap")
for pkt in packets:
```

```
    print(pkt.summary())
```

This code snippet lets the IDS process deauthentication packets, probe requests, unusual traffic spikes, MAC addresses and Packet counts. Deauthentication logic is added as:

```
from scapy.all import *

packets = rdpcap("capture.pcap")

deauth_count = 0

for pkt in packets:
    if pkt.haslayer(Dot11Deauth):
        deauth_count += 1
        print("Deauthentication packet detected!")

print("Total Deauth Packets:", deauth_count)
```

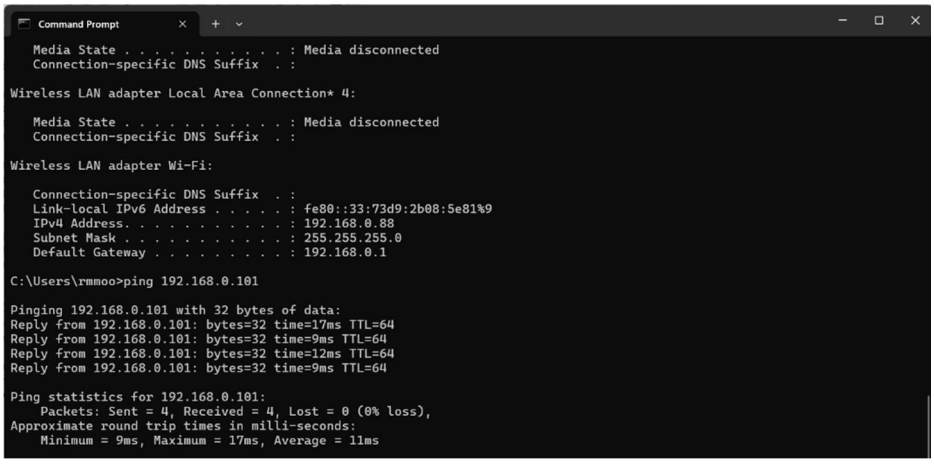
An Alert System is added by

```
from scapy.all import *

deauth_threshold = 5
count = 0

def detect(pkt):
    global count

    if pkt.haslayer(Dot11Deauth):
        count += 1
        print("[ALERT] Deauthentication Attack Detected!")

sniff(iface="Wi-Fi", prn=detect, store=0)
```

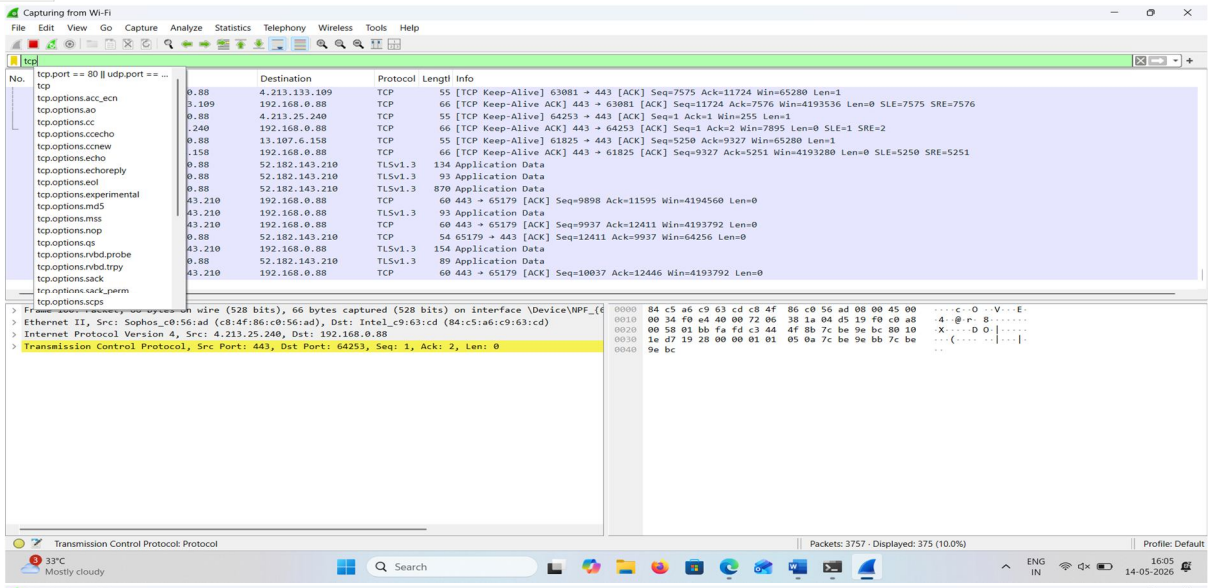
IDS can be connected to live traffic directly using:

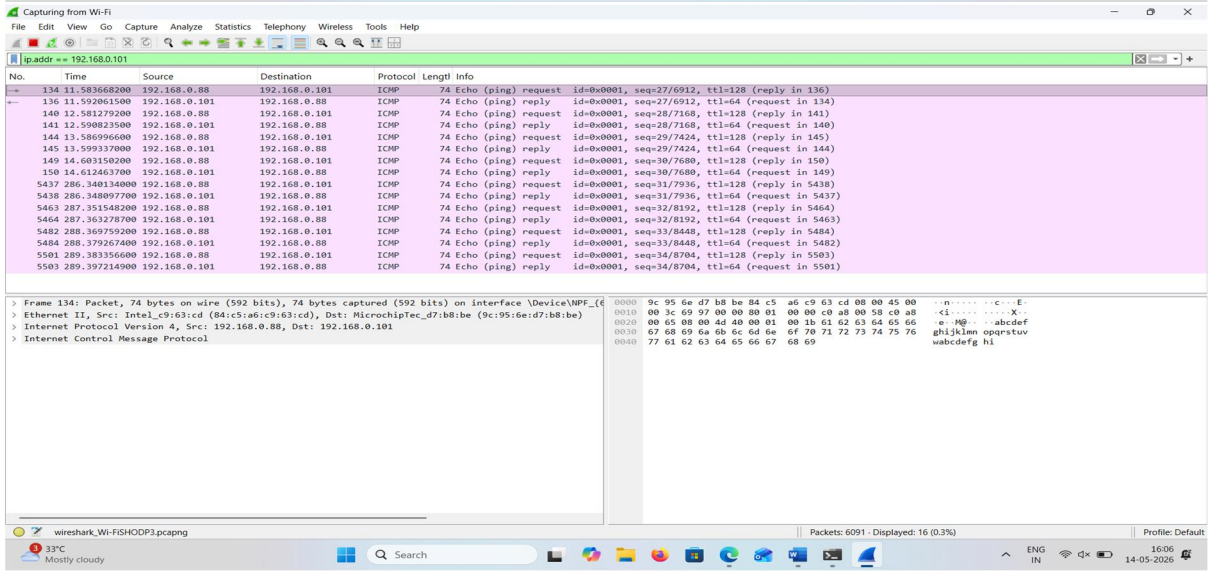
```
from scapy.all import *

def detect(pkt):
    print(pkt.summary())

sniff(iface="Wi-Fi", prn=detect, store=0)
```

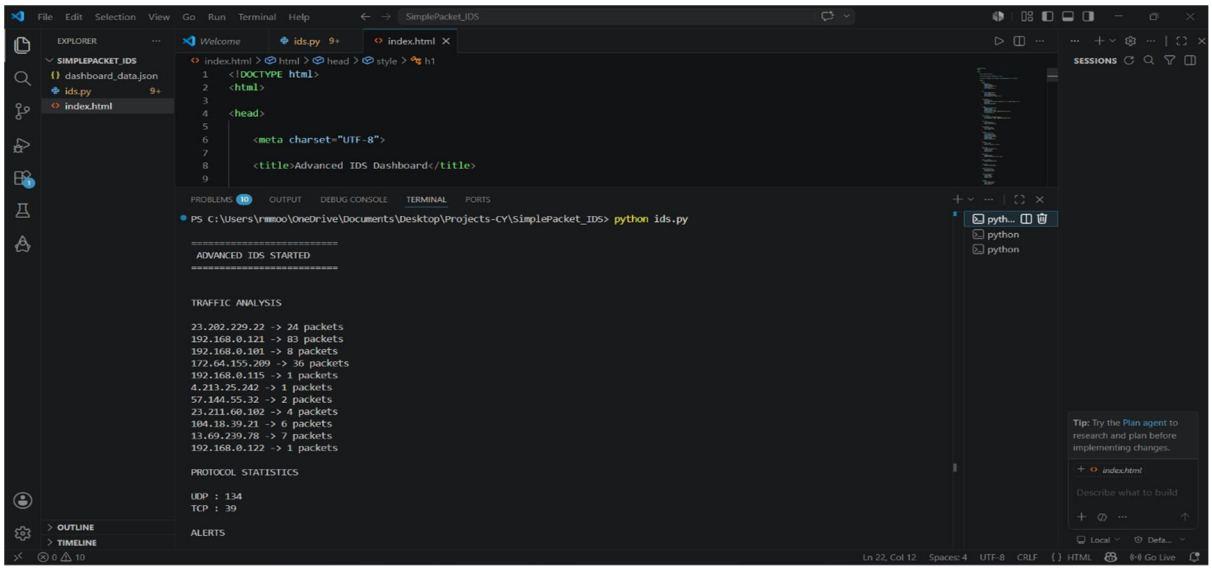
In my terminal if I run the file as python ids.py, it successfully reads the Wireshark packets.

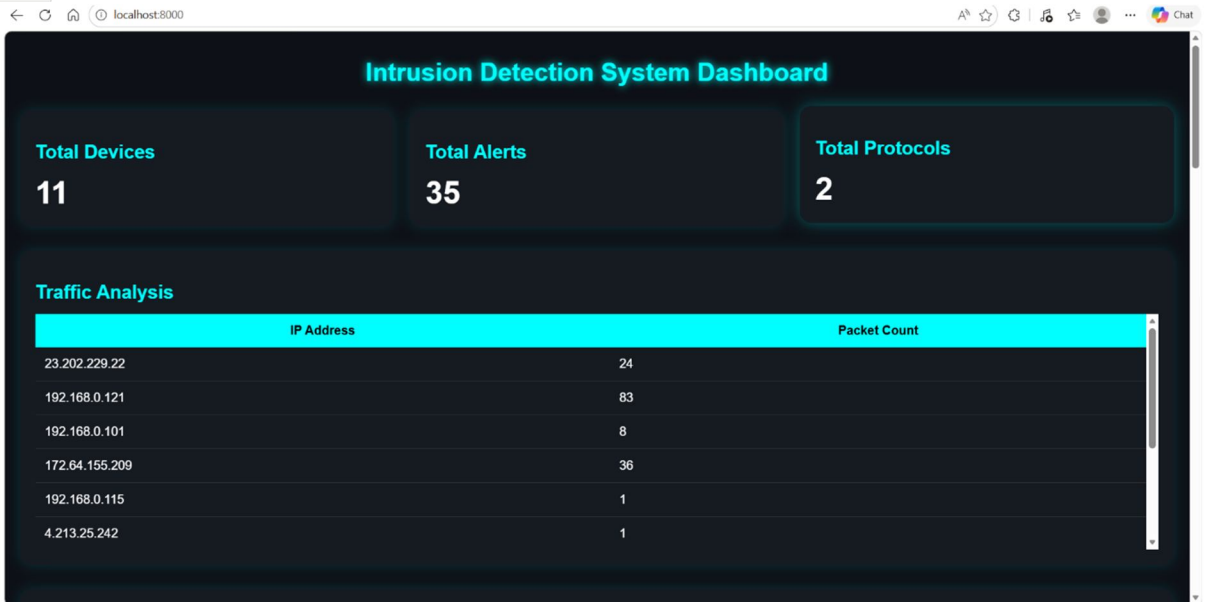


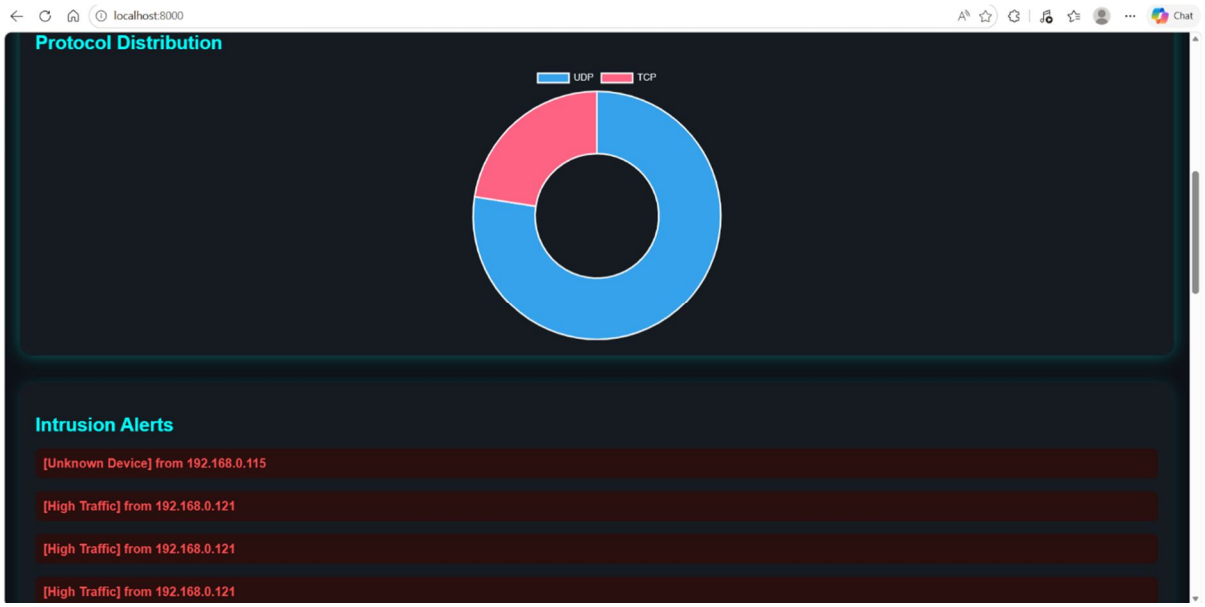
International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com











VI. CONCLUSION

Modern medical devices increasingly rely on wireless communication technologies for patient monitoring, data transmission, and remote healthcare applications. Many of these devices operate using WPA2-based wireless networks and embedded Linux platforms. Although WPA2 provides encryption for data transmission, medical devices remain vulnerable to various cybersecurity threats such as unauthorized access, traffic flooding, rogue devices, and network-based attacks. Since medical devices often handle sensitive patient information and support critical healthcare operations, any compromise in wireless communication may affect patient safety and system reliability.

The implementation of an Intrusion Detection System (IDS) significantly improves the security of medical devices by continuously monitoring network traffic and identifying suspicious activities in real time. Unlike traditional security mechanisms that only focus on prevention, an IDS actively analyzes packet behavior, protocol usage, traffic patterns, and communication anomalies to detect potential intrusions. This enables healthcare systems to identify abnormal network behavior before it causes disruption to medical services or compromises confidential patient data.

In the proposed system, the IDS performs packet inspection and traffic analysis using PCAP files captured through network monitoring tools such as Wireshark.

The IDS analyzes IP traffic, protocol distribution, packet frequency, and suspicious communication behavior. When abnormal traffic levels exceed predefined thresholds, the system generates intrusion alerts and displays them on the monitoring dashboard. This allows administrators to quickly identify potentially malicious devices or unusual communication activity within the healthcare network.

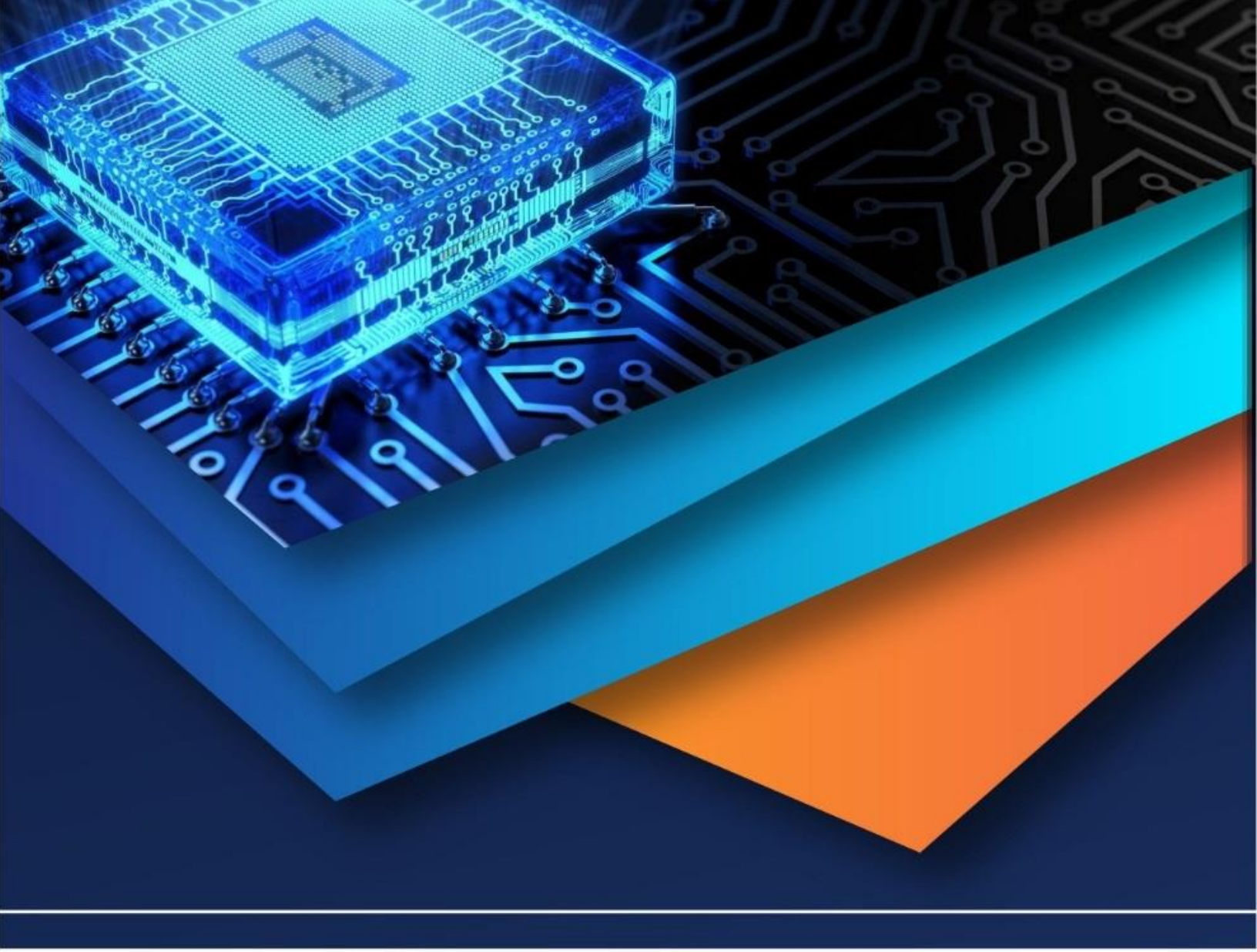
The IDS also improves visibility into medical device communication by identifying unknown devices connected to the network. In hospital environments, unauthorized wireless devices may introduce security risks such as rogue access points, malware propagation, or unauthorized access attempts. By monitoring IP addresses, packet counts, and protocol usage, the IDS helps administrators distinguish trusted medical devices from suspicious or unrecognized systems. This strengthens network access monitoring and reduces the risk of unauthorized communication within healthcare infrastructure.

Another important advantage of IDS implementation is its compatibility with legacy medical devices. Many healthcare systems continue to operate older embedded platforms that may not support advanced security features such as WPA3 or Protected Management Frames (PMF). The proposed IDS enhances cybersecurity without requiring hardware replacement or significant firmware modifications. As a software-based monitoring solution, the IDS can be integrated into existing WPA2-based medical environments with minimal impact on device performance and operational workflow.

The dashboard-based visualization system further enhances security management by providing real-time traffic analysis, protocol statistics, and intrusion alerts through an interactive monitoring interface. This enables healthcare administrators to observe network conditions continuously and respond rapidly to suspicious activity. The combination of packet monitoring, intrusion detection, and visual analytics improves the overall cybersecurity resilience of wireless medical systems while maintaining compatibility with existing infrastructure.

REFERENCES

[1] https://www.lifewire.com/what-is-wi-fi-2377430
[2] Taskin, M. (2008) WEP Post Processing Algorithm for Robust 802.11 WLAN Implementation. Science Direct: Computer Communication Journal 31, 3405-3409.
[3] Najar, Z. and Mir, R. (2021) Wi-Fi: WPA2 Security Vulnerability and Solutions. Wireless Engineering and Technology, **12**, 15-22. doi: 10.4236/wet.2021.122002.
[4] Vidhan Dilip Gambhire, Sushant Ramchandra Gade, Sandhya Kaprawan, Aniket Gupta University Department of Information & Technology (MSC. Cybersecurity), University of Mumbai. Wireless Wi-Fi Access Point Security: An Analysis of WPA, WPA2, WPS, and WPA3.
[5] A. S. Abrar, N. Patwari, and S. K. Kasera, "Quantifying Interference-Assisted Signal Strength Surveillance of Sound Vibrations," IEEE Transactions on Information Forensics and Security, vol. 16, p. 2018, Dec. 2020, doi: 10.1109/tifs.2020.3045316
[6] Zhang, Y. and Sampalli, S. (2010) Client- based Intrusion Prevention System for 802.11 Wireless LANs, WiMob2010. Proceedings of the 6ht International Conference IEEE 2010 on Wireless and Mobile Computing, Networking and Communication, Niagara Falls, 11-13 October 2010, 100-107.
[7] Lounis, K., Ding, S.H.H., Zulkernine, M. (2022). Cut It: Deauthentication Attacks on Protected Management Frames in WPA2 and WPA3. In: Aïmeur, E., Laurent, M., Yaich, R., Dupont, B., Garcia-Alfaro, J. (eds) Foundations and Practice of Security. FPS 2021. Lecture Notes in Computer Science, vol 13291. Springer, Cham. https://doi.org/10.1007/978-3-031-08147-7_16
[8] What is a Krack Attack? | Fortinet
[9] Alfaro, J.G., Cuppens, F., Cuppens-Boulahia, N. (2007). Management of Exceptions on Access Control Policies. In: Venter, H., Eloff, M., Labuschagne, L., Eloff, J., von Solms, R. (eds) New Approaches for Security, Privacy and Trust in Complex Environments. SEC 2007. IFIP International Federation for Information Processing, vol 232. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-72367-9_9
[10] Distributed WPA PSK strength auditor
[11] Shufeng Li, Mingyu Cai, Robert Edwards, Yao Sun, Libiao Jin, Research on encoding and decoding of non-binary polar codes over GF(2m), Digital Communications and Networks, Volume 8, Issue 3, 2022
[12] "[Front cover]," 2021 8th International Conference on Electrical and Electronics Engineering (ICEEE), Antalya, Turkey, 2021, pp. c1-c4, doi: 10.1109/ICEEE52452.2021.9415962.
[13] Manesh Thankappan, Helena Rifà-Pous, Carles Garrigues, Multi-Channel Man-in-the-Middle attacks against protected Wi-Fi networks: A state of the art review, Expert Systems with Applications, Volume 210, 2022
[14] FragAttacks: Security flaws in all Wi-Fi devices
[15] How to Build a Real-Time Intrusion Detection System with Python and Open-Source Libraries



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY