AUTOMATED WI-FI INTRUSION DETECTION TOOL ON 802.11 NETWORKS

Dimitris Koutras, Panos Dimitrellos, Panayiotis Kotzanikolaou, Christos Douligeris
Department of Informatics University of Piraeus, 80 Karaoli & Dimitriou st, Piraeus, Greece

NOTE: Corresponding author: Dimitris Koutras, dkoutras@unipi.gr

Abstract – *Wi-Fi networks enable user-friendly network connectivity in various environments, ranging from home to enterprise networks. However, vulnerabilities in Wi-Fi implementations may allow nearby adversaries to gain an initial foothold into a network, e.g., in order to attempt further network penetration. In this paper we propose a methodology for the detection of attacks originating from Wi-Fi networks, along with a Wi-Fi Network Intrusion Detection (Wi-Fi-NID) tool, developed to automate the detection of such attacks at 802.11 networks. In particular, Wi-Fi-NID has the ability to detect and trace possible illegal network scanning attacks, which originate from attacks at the Wi-Fi access layer. We extend our initial implementation to increase the efficiency of detection, based on mathematical and statistical function techniques. A penetration testing methodology is defined, in order to discover the environmental security characteristics, related with the current configuration of the devices connected to the 802.11 network. The methodology covers known Wi-Fi attacks such as de-authentication attacks, capturing and cracking WPA-WPA/2 handshake, captive portal and WPA attacks, mostly based on various open source software tools, custom tools, as well as on specialized hardware.*

Keywords – IEEE 802.11, network intrusion detection, Wi-Fi

NOTE: This work has been partly supported by the University of Piraeus Research Center.

1. INTRODUCTION

Wi-Fi networks, also known as IEEE 802.11, have become essential in our everyday lives, offering increased access and convenience. Yet, the nature of wireless communications makes them susceptible to interference, manipulation, or disruption by malicious individuals, creating significant security challenges. Proactively gathering and scrutinizing this data from a security perspective can lead to the prompt identification of suspicious activities and early resolution of potential configuration weaknesses. Because attackers often engage in this sort of network scanning as a preliminary step before launching an assault on the intended services or information, network administrators must prioritize the early detection and resolution of these vulnerabilities to prevent exploitation.

Packet capturing and analysis play a crucial role in network forensics [1]. Capturing packets in real time offers a comprehensive log of all communications, and this data can be analyzed immediately or reviewed later. Professionals in network security rely on pcap files to dissect network traffic, recreate network events, and pinpoint traffic origins and destinations. These files help ascertain traffic characteristics, including the protocols and applications in use. During incident management, pcap files are especially crucial because they present a factual and precise history of network activity for investigating incidents and detecting security infractions. Recognizing the critical role of pcap files in network forensics, is vital for security experts to be adept at capturing, preserving, and evaluating these files efficiently. While it's less typical to use this method for Wi-Fi network oversight, employing a similar approach can prove advantageous in scrutinizing and identifying suspicious behavior in 802.11 networks, thereby allowing an administrator to swiftly detect and address potential risks.

Wireless networks are particularly vulnerable to attacks, often because infiltrating them can be unexpectedly easy. Once attackers gain control over a wireless network, they gain an advantageous attack vector capability, i.e., adjacent network access, from which they can extend their attack to other systems within the internal network. One of the primary threats involves the theft of sensitive data. By breaching the network, cybercriminals can intercept any information being transmitted over the network, such as personal messages, financial information, or confidential business data. This information can be used for financial gain through selling the data on the dark web, identity theft, or even blackmailing purposes [2].

Gaining control also allows attackers to launch secondary attacks on connected or adjacent networks [3], often targeting more secure or high-value systems like corporate networks. These secondary attacks, originating from the compromised network, add layers of obfuscation, challenging cybersecurity professionals to trace back these incidents to their sources. The attackers' ongoing presence creates a continual security risk, as they can install backdoors or exploit existing vulnerabilities, ensuring persistent access and leaving the network open to additional future compromises. This sustained threat can have a

domino effect, weakening the broader security infrastructure connected to the initial breach.

To ensure the security and stability of network infrastructure, detecting and addressing harmful actions in wireless networks is crucial. The National Institute of Standards and Technology (NIST) recommends that organizations employ Intrusion Detection and Prevention Systems (IDPS) [4] to track and evaluate network traffic, as well as detect and respond to malicious activity [5]. By putting in place efficient detection strategies, organizations can diminish the threat of cyberattacks and protect their networks.

1.1 Motivation

As cyberattacks become more advanced and wireless networks remain exposed to risks, organizations are confronted with increasing security threats. Even with security measures like WPA2/WPA3 encryption in place, 802.11 networks are still prone to breaches. The ease with which attacks on Wi-Fi networks can be conducted, even by those with little technical know-how, is concerning. Thus, it's essential for both corporate environments and home Wi-Fi setups to enhance security by employing specific detection and response strategies. This effort complements proactive steps like using robust passwords, keeping security protocols updated, and consistently overseeing network activity.

In addition we try to enhance and innovate within the realm of attack detection methodologies. In an era where digital threats have become increasingly sophisticated, traditional defensive mechanisms often fall short in both speed and accuracy when identifying and neutralizing such threats. Our primary motivation in conducting this research is to bridge these gaps by developing superior and more efficient methods for detecting network intrusions and malicious activities. By doing so, we aim to refine the precision of attack detection systems, focusing on minimizing false positives without compromising sensitivity. This heightened accuracy ensures that security resources are not misdirected toward benign anomalies, which is crucial for maintaining optimal operational efficiency and allows for a more focused and effective response to actual threats. By reducing the incidence of false positives, we enhance the credibility of alarm systems and ensure that genuine threats receive immediate and undivided attention, thereby strengthening the overall security posture.

1.2 Contribution

In this paper, we define a methodology for the detection and attack propagation analysis of attacks originating from the Wi-Fi layer. In particular our main contributions are:

- We define a Wi-Fi-specific intrusion detection model, based on statistical metrics. The model uses the rolling median to dynamically calculate the appropriate threshold for detecting anomalies in Wi-Fi network traffic. Our dynamic detection model is correlated with standard detection techniques (e.g. pcap packet analysis and signature-based detection).
- The proposed model is capable of detecting Wi-Fi-specific attacks, not only attacks that originate from known Wi-Fi attack tools such as Aircrack-ng, Reaver, mdk4 etc, but also customized attacks. This is due to the fact that the proposed statistical model provides a systematic and highly reliable means of identifying anomalies that indicate potential security breaches.
- In addition to the Wi-Fi intrusion detection phase, the proposed methodology is able to evaluate the *propagation* of security attacks originating from Wi-Fi 802.11 networks and targeting internal networks beyond the Wi-Fi network layer. Based on a penetration testing phase, we identify the key security challenges faced by Wi-Fi 802.11 networks, in order to evaluate how Wi-Fi attack vectors can be extended and cascaded to network attack vectors. These include Denial of Service (DoS) attacks on the WPA2 handshake, social engineering and DoS attacks using a fake captive portal, exploiting the WPS at the access point and de-authentication attacks.

To evaluate the proposed methodology, we have implemented Wi-Fi-NID[1], a Wi-Fi network intrusion detection tool that can detect and respond to security incidents. Wi-Fi-NID offers an innovative approach to detecting malicious activity in Wi-Fi networks, by focusing on Wi-Fi-specific attack features to identify attacks that originate from the 802.11 layer. As Wi-Fi-NID operates at the edge of the Wi-Fi network, it can be easily integrated as an add-on security mechanism and may be complementary to general IDS solutions that do not focus at the Wi-Fi layer. By providing a level of protection critical to maintaining the security and integrity of Wi-Fi networks, Wi-Fi-NID represents a valuable addition to any organization's network security toolkit. Finally we present the validation of Wi-Fi-NID based on various test scenarios. During our validation, we tested different approaches that exploit various vulnerabilities in WPA2, which were successfully identified and analyzed.

1.3 Paper structure

In Section 2 we review related work. In Section 3 we outline the suggested methodology for conducting Wi-Fi penetration tests, while in Section 4 we describe Wi-Fi-NID, an automated tool that implements the proposed methodology. In Section 5 we confirm the effectiveness of our model by showcasing outcomes for each attribute of the

[1] https://github.com/panosdimitrellos/Wi-Fi-NID

tool. Finally, Section 6 concludes this paper with a discussion about future plans.

2. RELATED WORK

2.1 Wi-Fi attack detection tools

Several intrusion or attack detection tools can be found in the literature. In [6] an IDS is proposed, which is highly effective against WPA2/3 de-authentication attacks. The proposed system has some level of automation, but not against various Wi-Fi attacks, while it is not able to identify attacks such as WPS null pin attack. Chen et al. [7] suggest a method for operator networks to detect phishing attacks. But none of the aforementioned solutions concentrate on threat analysis for Wi-Fi networks, either alone or in conjunction with automated pcap file analysis tools.

In [8] authors propose a lightweight and efficient DDoS attack detection approach using change point analysis, which demonstrates high detection rates and linear complexity, making it suitable for WSNs. They use change point detectors to monitor anomalies in two metrics: the data packets' delivery rate and the control packets' overhead.

Wi-Fi-IDS[2] is a tool that can sniff Wi-Fi traffic for malicious activity, like WEP, WPA, and WPS packet attacks, but it is unable to identify some of the attacks that the Mdk4 tool[3] can detect, such as WPA2 and null pin attacks.

In [9] authors include variations in the minimum DDoS attack rates that cause disruptions in different IoT smart home devices, the impact of Wi-Fi group key updating on DDoS attacks, and the factors affecting the energy consumption of victim devices during attacks. This research aims to enhance the understanding of IoT device vulnerabilities in smart home environments and lay the groundwork for future defense solutions.

The work of [10] introduces "EvilScout," a framework that uses IP-prefix distribution information from the legitimate Access Points (APs) to detect evil twins. EvilScout leverages software-defined networking to detect evil twins without the need for extra hardware or modifications to the AP or client.

The work of [11] categorizes association attacks on Wi-Fi clients, based on the specific network manager features that are exploited by each attack, using the Wi-Fiphisher tool. The paper explore various strategies for increasing the success rate of these attacks and assess their impact on new security protocols like WPA3, Wi-Fi Enhanced Open, and Easy Connect.

In [12], the authors address the vulnerability of WLAN-based localization systems to location spoofing attacks, which present major privacy concerns in Mobile Social Network Services (MSNS). They introduce a privacy attack model based on spoofing in MSNS and propose a defense mechanism using Wi-Fi-hotspot tags (base-station tags, BS tags) for authenticating spatial-temporal properties of geolocations.

In [13] the authors propose an IoT-based method to detect and prevent fake access point attacks in Wi-Fi networks. Utilizing a single board computer and a wireless antenna with a "Soft AP" feature, they conducted air scans to identify and mitigate fake AP broadcasts.

In our preliminary work [14], an initial version of our detection framework presented in this paper for Wi-Fi attacks is presented. However, in [14] the application of a static threshold in attack detection is applied, encountering significant challenges in the form of pervasive false positives. In the initial version, the system was capable of detecting only those Wi-Fi attack scenarios where the number of packets sent by the attacker exceeded a predetermined static threshold. This observation led to the consideration and subsequent adoption of a dynamic threshold.

2.1.1 Wi-Fi ML attack detection tools

We examine indicative examples of Wi-Fi detection tools that incorporate Machine Learning (ML) techniques within their methodologies. The work of [15] explores the potential of using network profiling and ML to secure IoT against cyberattacks. The proposed anomaly-based intrusion detection solution dynamically and actively profiles and monitors all networked devices for the detection of IoT device tampering attempts, as well as suspicious network transactions. Raw traffic is also passed on to the machine learning classifier for examination and identification of potential attacks.

In [16], an ML-based Wireless Intrusion Detection System (WIDS) is developed to efficiently detect attacks on wireless networks, using attribute selection methods to improve accuracy and speed. The system's performance, evaluated on the Aegean wireless intrusion dataset using tools like Weka, Rstudio, and Anaconda Navigator Python, demonstrates the effectiveness of the chosen ML algorithm.

In Table 1, we present a comprehensive comparison of the various Wi-Fi intrusion detection systems previously discussed. This comparison focuses on four critical dimensions: the number of attacks detectable by each system, the capability to identify attacks originating from custom tools, the implementation of dynamic methodologies such as dynamic detection thresholds, and the effectiveness in evaluating and elucidating attack propagation. The criteria are chosen to highlight the strengths and limitations of each system in a practical security environment.

- Number of detectable attacks: This criterion evaluates the breadth of the attack spectrum that each IDS can identify. A higher number of detectable attacks indicates a more versatile and comprehensive detection capability, which is crucial for effective network security.

[2] https://en.kali.tools/?p=83

[3] https://github.com/aircrack-ng/mdk4

Table 1 – Related work - comparison

Paper	Number of attacks covered	Dynamic Threshold	Detection of attacks from custom tools	Evaluation of attack propagation
Current work	up to 25	Yes	Yes	Yes
[14]	up to 20	No (static threshold)	No	Yes
[7]	up to 10	No (avg Function)	Not mentioned	No
[17]	up to 10	No (Frames/sec)	No	No
[8]	up to 10(DoS)	No (time besed function)	Not mentioned	No
[10]	up to 10	No	Not mentioned	No
[15]	up to 15	Dynamic technique	Yes	No
[16]	up to 20	Dynamic technique	Not mentioned	No
[18]	up to 20	Dynamic technique	No	Yes (partially)
[12]	up to 10	N/A	No	Yes (partially)
[13]	up to 5	No	No	No

- Detection of attacks from custom attack tools: Given the evolving nature of cyberthreats, it is imperative for an IDS to recognize attacks that are not part of standard threat databases, particularly those originating from custom, previously unseen tools. This criterion assesses each system's adaptability and resilience against novel threats.
- Use of dynamic detection methodologies: The implementation of dynamic methodologies, such as dynamic detection thresholds, plays a vital role in the adaptability of an IDS. These methodologies enable the system to adjust its detection parameters in real time, enhancing its effectiveness in diverse and changing network environments.
- Evaluation of attack propagation: This aspect examines each system's capability to not only detect but also evaluate the propagation and potential impact of an attack, such as extending Wi-Fi attack vectors to network attack vectors. Understanding attack propagation is essential for developing effective mitigation strategies and for providing insights into the security posture of the network.

2.2 Penetration testing methodologies for 802.11 wireless networks

Several general methodologies for penetration testing exist in the literature, e.g. [19, 20]. The first strategy relies on using different tools, like IDS for forensic analysis. The second strategy emphasizes the value of employing red teaming resources, which is noteworthy and pertinent in our situation. The paper of [21] provides a methodology for a number of network attacks, including packet sniffing, DoS attacks, and unauthorised access attacks, that is extremely similar to what we need. Matthew et al. [22] carries out a penetration test using a variety of attacks, but only with the help of a sophisticated port scanner.Building upon the methodologies presented in [19] and [20], our approach incorporates, we will export and analyze forensic data from pcap files using a variety of red teaming tools and specialized hardware. We will also concentrate on attacks that are pertinent to Wi-Fi networks.

2.3 Malicious MAC address-related mechanisms

Finding malevolent MAC addresses could be a crucial first step in stopping 802.11 layer attacks. The literature has identified a number of relevant mechanisms. For example, Girdler et al. [23] present an IDPS system that focuses on malicious MAC addresses associated with ARP spoofing attacks. Yaibuates et al. [24] try to identify malicious requests for IP addresses via DHCP, and combine the ICMP and ARP protocols. The paper of Hsu et al. [25] suggests a method that uses a range of reverse traceroute data gathered by a remote server to identify the existence of a malicious rogue AP.

In Anathi et al. [26], in order to provide a thorough analysis of its efficacy in identifying and mitigating threats, a network localization-based approach is presented that is supplemented by the use of port scanning, OS fingerprinting, and route tracing algorithms. Finally, Wang et al. [27] present an Unmanned Aerial Vehicle (UAV)-based rogue Wi-Fi access point recognition system. This system makes use of Software-Defined Radio's (SDR) wireless analysis capabilities, as well as the remarkable mobility capabilities of UAVs.

2.4 Pcap file inspection

In the literature, several pieces of work focus on packet capture and pcap file analysis. For instance [28] propose a technique for analyzing pcap files that focuses on minimizing the duration of the parsing process. Kismet[4] is a packet sniffing application with IDS features that supports pcap files, and it mainly makes use of the device's network card. It does not have automated capabilities for analyzing pcap files to find Wi-Fi assaults, despite the fact that it can identify active wireless spying programs. Deri et al. [29] uses the packet header and the payload from network analysis traffic packets in order to detect the application layer protocol employed during the transaction of information. Contrary to our approach, which leverages the entirety of information provided by a pcap file. The Song et al. [30] approach presents noteworthy aspects worth considering. They try to detect network anomalies in the system behavior. But in their case, they use standard intrusion detection and prevention systems. Instead, in our case we try to make our own intrusion detection system with a very specific target. Another proposal entails sourcing information from proxy logs instead of pcap files, utilizing the pcap files as supplementary data for cross-validation and timeline analysis [31].

3. THE PROPOSED METHODOLOGY

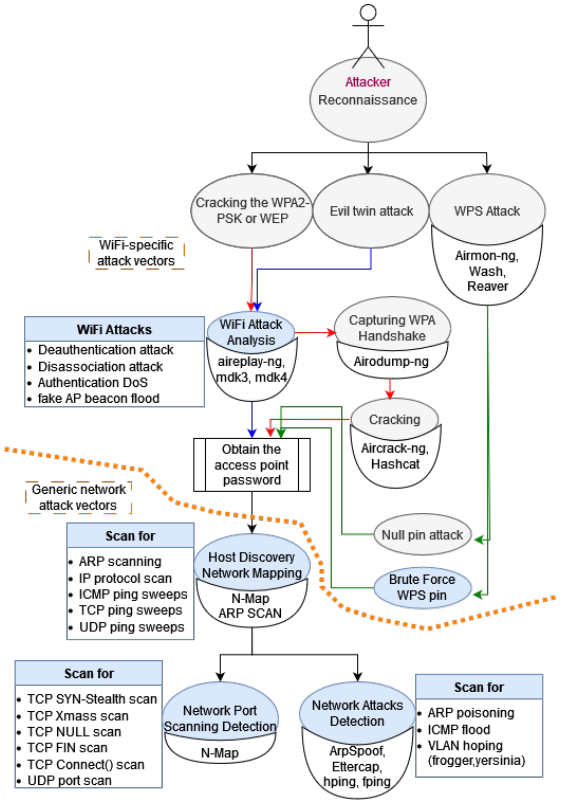


Fig. 1 – The proposed methodology: Wi-Fi-specific attack vectors are captured and correlated with attack indicators identified in internal hosts

[4]https://www.kali.org/tools/kismet/

The main goal of the proposed methodology is to detect and analyse network attacks originating from the Wi-Fi layer and propagating to other internal networks. The proposed methology involves two main phases, as shown in Fig. 1. In the first phase, we utilize detection techniques based on dynamic thresholds for detecting anomalies in Wi-Fi network traffic. In the second phase, we combine the detection results identified in the first phase, in order to evaluate the propagation of attacks originating from the Wi-Fi networks to internal networks, based on penetration testing techniques.

3.1 Phase 1 - Analysis of Wi-Fi-specific attack vectors

The initial phase of our approach focuses on the early detection of potential Wi-Fi-specific attack vectors that aim to infiltrate wireless networks. These vectors include, but are not limited to, more advanced techniques and cracking attempts directed at encryption protocols like WPA2, PSK, or WEP, as well as complicated tactics like evil twin attacks, de-authentication attacks, and exploitation of vulnerabilities within the WPS protocol.

At this point, our methodology can engage penetration testing approaches, in order to test the responses of attacks starting from the Wi-Fi vector and targeting the inner systems. As shown in Fig. 1, we define three different flows (denoted with three different colors) starting from three different attacks. For the implementation of these three attack flows, from the initial attack to the penetration of the system, various custom and non-custom attack tools were used. In the first attack flow (red color), encryption cracking constitutes a threat to wireless security, as breaching WPA2, WPA3, PSK, or WEP barriers grant attackers access to confidential information transmitted over the network. In the second attack flow (blue color), by using the evil twin attack, attackers may set up a counterfeit access point mimicking a legitimate one, deceiving users into establishing a connection. This deceptive strategy not only enables the interception of user traffic but also paves the way for additional, often more harmful, attacks. Finally, in the third attack flow (green color) attacks in the WPS protocol can be exploited to allow attackers to retrieve the network's WPA2-PSK passphrase. These methodologies, while straightforward in execution, represent substantial security risks. In scenarios involving encryption cracking or the deployment of an evil twin attack, the adversary commonly resorts to de-authentication or disassociation attacks. These are preliminary steps to disconnect users from the legitimate network, making it easier to retrieve the access point's password through methods such as handshake capturing. Furthermore, the attacker might execute a Denial of Service (DoS) attack, inundating the network with authentication packets or creating a barrage of fake AP beacons, thereby disrupting normal network operations.

We aim to intercept malicious communications at the critical intersection before an attacker, leveraging any of the aforementioned Wi-Fi-specific techniques, succeeds in acquiring the Wi-Fi password and intruding into the wireless network. This preventive stance on early-stage intrusion attempts is foundational to our comprehensive security framework, serving as a pre-emptive measure to protect sensitive data and maintain network integrity. Our methodology underscores the importance of not only recognizing the symptoms of an attack but also intervening decisively before the security breach matures, thereby preserving the sanctity of the wireless environment.

3.1.1 Formulating detection of Wi-Fi-specific attacks

In our study, we have established a methodology for detecting Wi-Fi-specific attacks using the rolling median. This approach is particularly effective in addressing the substantial traffic variations characteristic of Wi-Fi layer attacks. The rolling median, as opposed to the mean, demonstrates a robust resistance to extreme values. It's imperative to understand that the median, being the central value in a sorted data array, remains unaffected by the magnitude of outliers, relying solely on their relative placement. Conversely, the mean is highly susceptible to such extremes, with a single anomalous value capable of significantly altering its calculation.

Crucially, the median maintains consistency amidst varied data points, reflecting the core tendency of the dataset without distortion from extreme values. This attribute renders the median a more reliable indicator of the overall data trend, ensuring that the analysis is not disproportionately influenced by outliers.

Given that the rolling median remains relatively stable even in the presence of outliers, it becomes a preferred choice for detecting anomalies or outliers, as in this case. The rolling median essentially sets a dynamic threshold, which allows the detection of outliers in a dataset. We achieve this by checking if each element in the list of, for example de-authentication packet counts per second, exceeds the calculated threshold. When an element surpasses this threshold, it is considered an outlier and is categorized into a new list of outliers for further analysis. Non-outliers are marked with a value of 0, while outliers retain their original values. This way, we filter the data to identify values that deviate from the desired threshold. Essentially, the rolling median's ability to ignore very high or low points and focus on the more consistent, central data makes it a stronger tool for identifying odd data points that might be worth our attention.

Once an attack is spotted, we take a closer look at the CSV file (it is obtained via a custom script from the pcap file) to figure out who the targets are. From our tests with regular traffic and during different attacks, we noticed that de-authentication packets aren't common and show up more frequently when an attack is happening, causing a big jump in their numbers. So, to find the targets, we check which MAC addresses are sending out the most de-authentication packets and who they're sending them to the most. This helps us see the main players in the attack, so the 'Client' is sending the packets and the 'Access Point' receiving them. We also make a note of the packets going to and from these targets to get the full picture of the attack.

3.1.2 Statistical model

The Rolling Median (RM)[5] is a statistical measure representing the median of a group of numbers. In this instance, the rolling median calculates the dynamic threshold for detecting anomalies in network traffic, such as a de-authentication attack. It is represented as:

$$RM(X, w, t) = Median(X[t - w/2 : t + w/2])$$

Where:
X : represents the dataset or time series.
w : is the window size, which is typically an odd integer to ensure a clear center.
t : is the point in time or index where you want to calculate the rolling median.
$[t-w/2 : t+w/2])$: represents the window of data points, including t and containing w data points. The median is calculated within this window.

The expression for the median of a dataset with an even number of data points can be represented as follows:

$$Median = \frac{X_{\frac{n}{2}} + X_{\frac{n}{2}+1}}{2}$$

Where:
Median : represents the median value.
X : represents the dataset.
n : is the total number of data points in the dataset.
$X_{\frac{n}{2}}$: represents one of the middle data points when the data is sorted.
$X_{\frac{n}{2}+1}$: represents the other middle data point when the data is sorted.

To calculate the rolling median, a moving window of a specified size (e.g., 10 seconds) is applied to find the median of packet counts within each window. This offers a smooth representation of de-authentication packet counts over time, simplifying the identification of data trends and patterns.

The dynamic threshold T is derived from the Rolling Me-

[5] https://pandas.pydata.org/docs/reference/api/pandas.core.window.rolling.Rolling.median.html

dian (RM) of the dataset. It is calculated using the mean and standard deviation of the rolling median, with the threshold computed as the mean plus one standard deviation. The general expression of the threshold is as follows:

$$T(RM) = \mu(RM) + \sigma(RM)$$

Where:
$T(RM)$: represents the threshold value for the rolling median.
$\mu(RM)$: represents the mean (average) of the rolling median.
$\sigma(RM)$: represents the standard deviation of the rolling median.

The general expression for the mean (μ) of the Rolling Median (RM) is as follows:

$$\mu(RM) = \frac{1}{n}\sum_{i=1}^{n} RM_i$$

Where:
RM : represents the rolling median dataset.
n : is the total number of data points in the rolling median dataset.
RM_i : represents an individual data point within the rolling median dataset, where i ranges from 1 to n.
$\sum_{i=1}^{n}$: denotes the summation of all the data points from $i = 1$ to $i = n$.

The general expression for the standard deviation (σ) of the Rolling Median (RM) is as follows:

$$\sigma(RM) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(RM_i - \mu(RM))^2}$$

Where:
$\sigma(RM)$: represents the standard deviation of the rolling median.
RM : represents the rolling median dataset.
n : is the total number of data points in the rolling median dataset.
RM_i : represents an individual data point within the rolling median dataset, where i ranges from 1 to n.
$\mu(RM)$: is the mean of the rolling median, as previously defined.
$\sqrt{...}$: denotes the square root of the expression inside, which provides the standard deviation.

This approach ensures dynamic threshold adjustment based on current packet counts, enabling detection to adapt to changing network conditions and provide precise results.

The value of the threshold **(T)** is derived from the function **T(RM) = μ(RM) + σ(RM)***, where it *dynamically* depends on and is shaped by the value of the Rolling Median **(RM)**. It is calculated from the mean (μ) and standard deviation (σ) values of each capture provided as input.

The rolling median value **RM(X, w, t) = Median(X[t - w/2 : t + w/2])** depends on:

- X, representing the time series the captured traffic at our disposal,
- w, which is the size of the rolling window we define as the desired detection window in the the captured traffic,
- and the value t, which is the point in time or index where the rolling median is calculated.

The only value we control is the size of the time window **w** that we want to monitor and use to partition the captured traffic. This gives us the ability to achieve better results by breaking down the captured traffic into smaller pieces, allowing us to take more samples from the captured traffic and avoid false positives or true negatives due to sparse non-ordinary values. Essentially, the rolling median's ability to ignore very high or low points and focus on the more consistent, central data makes it a stronger tool for identifying odd data points that might be worth our attention.

3.1.3 Visualization

To visualize the rolling median and dynamic threshold, a scatter plot of packet counts over time is generated, shown in Fig. 2. We plotted a scatter plot of de-authentication packets per second over the period of time. When an attack is detected, a label reading "Attack Detected" is displayed, indicating an anomaly in network traffic along with a red line representing the dynamic threshold and (x) points indicating the malicious traffic of de-authentication packets. When no attack is detected and normal network traffic is observed, a label reading "Normal Traffic" is displayed, signifying the absence of anomalies in the network traffic.The rolling median is shown as a dashed red line and acts as the dynamic threshold for detecting network traffic anomalies. When an attack is detected, an 'Attack Detected' label is displayed, indicating an anomaly in the network traffic. When no attack is detected and normal network traffic is observed, a label reading "Normal Traffic" is displayed (Fig. 3), signifying the absence of anomalies in the network traffic.

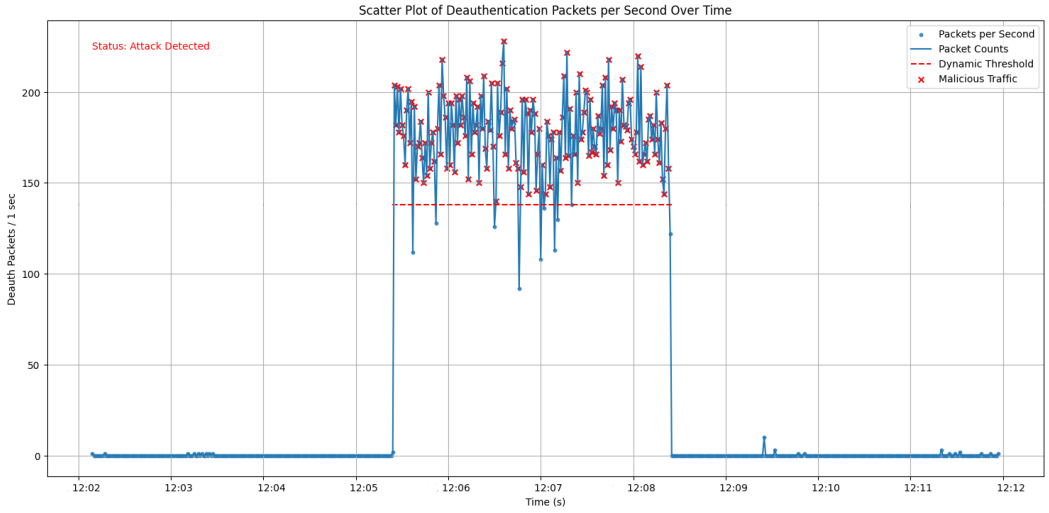


Fig. 2 – De-authentication attack detection

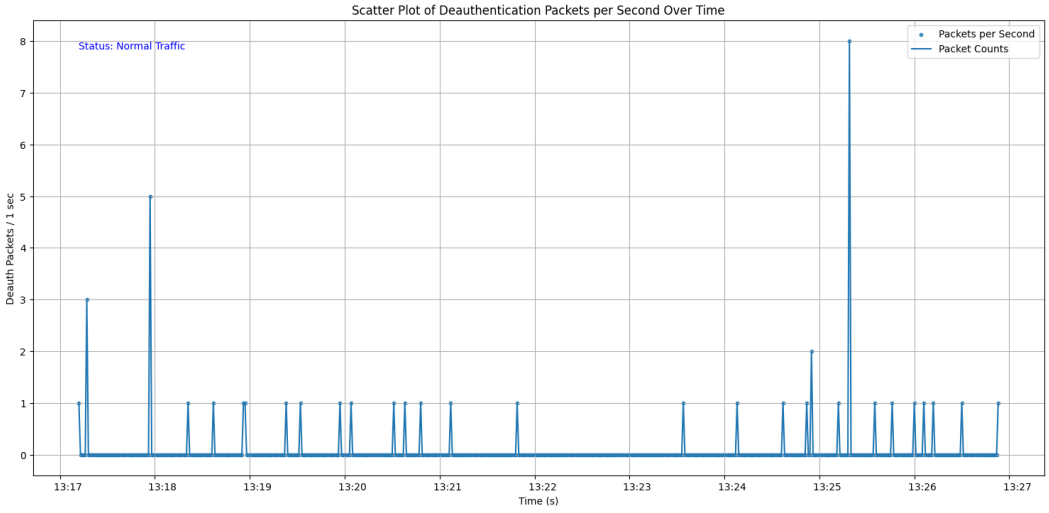


Fig. 3 – Normal packet transactions

3.2 Phase 2 - Analysis of generic network attack vectors

After successfully evading the Wi-Fi authentication layer, an attacker will try to move horizontally deeper into the network in the ensuing phase (shown in Fig. 1 by the blue, green, and red arrows, respectively). Assuming that the primary objective of preventing Wi-Fi intrusion has not been achieved, our methodology applies a second line of defense by identifying potential attacker attempts that are already connected to the Wi-Fi network. This is done by scanning attempts made within the internal Wi-Fi network in order to learn more about the network.

Although network security policies are typically set up to examine intrusion attacks coming from the external network, it is a common mistake to underestimate such attacks from hosts within the network. To identify post-Wi-Fi attack exploitation, the suggested methodology combines host identifying and network mapping attempts coming from the internal network with indicators of interference with the Wi-Fi network. While network mapping produces a visual depiction of the topology of a network, host discovery deals with finding active hosts on a network. Port scanning is used to find open ports and services on hosts. Together, these methods offer a thorough picture of network activity, enabling network managers to take preventative action to safeguard their infrastructure. In the proposed methodology, when malicious traffic is detected at more than one points, an attempt to construct an "activity chain" of the attacker is made, in order to capture the potential attack pattern starting from outside and continuing within the network.

Furthermore, a crucial aspect shaping our methodology pertains to the manner in which we identify each type of attack. As previously mentioned, we employ mathematical and statistical algorithms to identify Wi-Fi-related attacks, taking into account metrics like packet count (de-authentication and disassociation attacks). Nevertheless, certain attacks aren't reliant on packet quantities alone. These sub-attacks necessitate the utilization of data extracted from the pcap file for their detection. It's worth noting that each attack relies on distinct data sources, obtained through packet disassembly and protocol filtering. Additionally, in specific scenarios, custom scripts are employed for various purposes, always in conjunction with packet and protocol information.

4. WI-FI-NID IMPLEMENTATION

Wi-Fi-NID implements the attack detection techniques described in Section 3, and can be easily integrated into corporate or home Wi-Fi networks to minimize the risk of security incidents originating from vulnerable Wi-Fi networks.

The goal of Wi-Fi-NID is to detect common Wi-Fi-specific attacks early and to correlate them with suspicious network attacks originating from hosts directly or indirectly connected to Wi-Fi networks. Wi-Fi-NID implements the proposed methodology to detect: (a) Wi-Fi-based attacks in 802.11 networks and (b) deeper network intrusion attempts that originate from 802.11 networks.

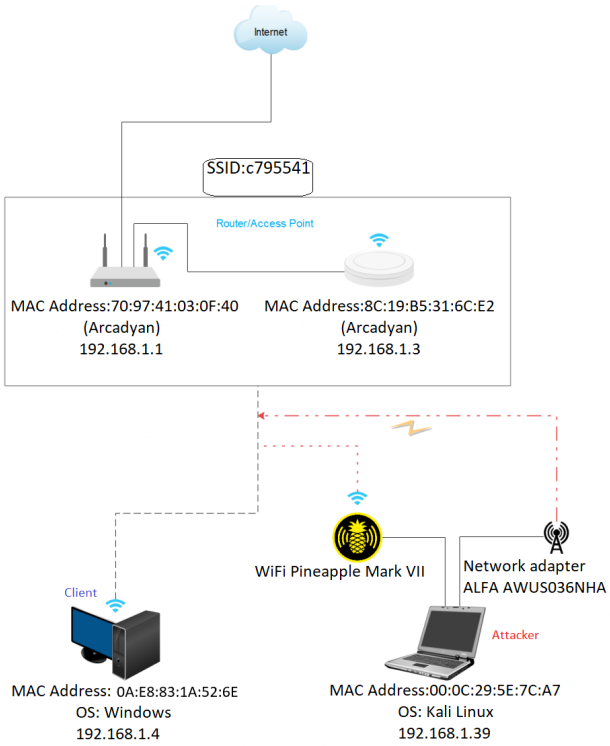


Fig. 4 – Experimental topology

4.1 Wi-Fi-based attacks in 802.11 networks

Wi-Fi-NID can detect all common Wi-Fi attack flows. Attack capture works as follows:

- De-authentication attack: This kind of attack disconnects authentic users from the network by sending forged de-authentication packets to a wireless access point. Tracing de-authentication attacks is based on the dynamic threshold detection method, described in Section 3.1, to dynamically trace the de-authentication packets.
- Disassociation attack: Detection is similar to the one described in the de-authentication attack, but in this case using disassociation packets [11].
- Authentication DoS: This kind of attack overloads an access point with requests for authentication, making it unusable for authorized users. We look for a high volume of authentication packets originating from different MAC addresses in order to identify this attack.
- Fake AP beacon flood: We track if there are a lot of random beacons in a wireless network because this kind of attack floods it with phony access point beacons.

4.1.1 *Wi-Fi traffic filtering and preparation*

During the first phase of the methodology, the captured network traffic, stored in pcap form, undergoes a filtration process to isolate the packets relevant to Wi-Fi layer attacks and to record their specific timestamps.

After filtration, the data is reformatted into a CSV file. This transformation allows for the establishment of a DataFrame, which forms the basis for the current analysis. For this process, we utilize the Pandas library, known for its efficiency in handling large datasets. We proceed by loading the CSV file, thereby creating a DataFrame replete with relevant information.

For the accuracy of time-based analysis, we convert the 'Time' column entries in the DataFrame into datetime objects. This conversion is preparatory to the segmentation of data into one-second intervals, a crucial step that enables the quantification of Wi-Fi de-authentication and disassociation packets on a per-second basis. As part of this detailed process, we extract essential parameters, notably the specific time segments and the corresponding counts of packets. These values are pivotal for generating scatter plots and identifying patterns in the network traffic. The preparation of the captured traffic strengthens our analysis by grounding it in concrete data. As a result, the detection of de-authentication packets becomes a more streamlined and efficient process, increasing the overall reliability and integrity of our network security measures.

4.2 Deeper network intrusion detection

Wi-Fi-NID also implements detection techniques for the identification of generic network attack vectors, which may be related with the identified Wi-Fi attacks.

Network mapping is one of the first actions an attacker will take to escalate their attack after gaining access to the network, with the goal of discovering other devices on the network and the services running on them to further penetrate and pursue the target. At this layer, Wi-Fi-NID looks for possible illegal network scanning (e.g. Nmap[6] scans or relevant tools). We have included various parameters and data resulting from the packets such as, the [destination] port state, flags like RST, FIN, URG, PSH and the size of the packet. Typical network scanning attempts such as TCP SYN Scan, stealth scan and Xmass scan are captured and analyzed.

Wi-Fi-NID is written as an automated script that combines Bash Scripting language with Python scripts. The tool has been tested on Kali Linux (Release: 2022.3). The key utility tool used to implement Wi-Fi-NID is TShark. TShark[7] is a network protocol analyser that is used in Command Line Interface (CLI). It allows us to capture packet data from a live network or read packets from a previously saved log file, either by printing a decoded form of those packets to standard output or by writing the packets to a file. In addition, TShark is able to detect, read and write the same download files that are supported by Wireshark. Using the proper parameters and filters, the relevant security events were recorded and displayed. Other command line utilities that have been used to filter the output of TShark include awk[8]. Awk is a tool that allows a programmer to write tiny but efficient programs in the form of statements that specify text patterns to search for on each line of a graph, and the action to be taken if a match is found within a line.

In other words, the content of packets originating from different malicious actions has been studied at a high level of granularity compared to packets corresponding to actual network traffic. The idea is to identify these different or malformed packets and use their content to identify the threat.

5. VALIDATION

A variety of test scenarios have been put into place and tested on testbed Wi-Fi networks made up of access points made by various manufacturers in order to validate the suggested tool. This implementation made use of specialized hardware, as well as a number of software tools created expressly to break into 802.11 networks. Numerous attacks were carried out in various ways with various pieces of equipment. The devices and topology utilized to execute a test case are depicted in Fig. 4. The topology's technical specifications are listed below.

A number of software tools, such as the password-cracking program Hashcat and the Aircrack-ng suite, which consists of Airmon-ng, Airodump-ng, and Aireplay-ng, can be used to record and crack WPA/WPA2 handshakes. An Alfa AWUS036NHA Wi-Fi adapter was used to apply the attack tools, capture packets, and carry out the attack. We also used a Wi-Fi Pineapple Mark VII, a specialized tool for evaluating Wi-Fi security, to carry out the attacks. The WPA/WPA2 handshake was captured and cracked using Wi-Fi Pinapple, along with the Pineapple's password cracking tools and the Recon Scan module for network scanning.

Tools like Aircrack-ng suite, Wash, and Reaver, all of which are frequently found in Kali Linux OS, can be used to carry out the WPS attack using the Null PIN method. It is important to note that every device tested with WPS enabled was completely vulnerable to the WPS attack using the Null PIN. In addition, it was executed quickly everywhere even though the password was complicated.

The MDK4 module and the Aircrack-ng suite, which includes Airmon-ng, Airodump-ng, and Aireplay-ng, can be used to launch Denial of Service (DoS) attacks. Ethical

[6] https://nmap.org/

[7] https://www.wireshark.org/docs/man-pages/tshark.html

[8] https://www.gnu.org/software/gawk/manual/gawk.html

Table 2 – Validation table - "Wi-Fi-NID" results

Feature	Categories	Packets	Source	Destination	OBSERVATION
Wi-Fi - specific Attacks	de-authentication attack	588 529 1185	14:ab:c5:c4:66:2f 8c:19:b5:31:6c:e4 8c:19:b5:31:6c:e4	8c:19:b5:31:6c:e4 14:ab:c5:c4:66:2f 8c:19:b5:31:6c:e4	Possible de-authentication DoS attack detected involving MAC address 8c:19:b5:31:6c:e4 due to high number of Deuthentication packets. If this happens in a high volume and in a small period of time,that indicates a high possibility of such an attack.
	Disassociation attack	590 588 1190	14:ab:c5:c4:66:2f 8c:19:b5:31:6c:e4 8c:19:b5:31:6c:e4	8c:19:b5:31:6c:e4 14:ab:c5:c4:66:2f 8c:19:b5:31:6c:e4	Possible Disassociation DoS attack detected involving MAC address 8c:19:b5:31:6c:e4 due to high number of Disassociation packets. If this happens in a high volume and in a small period of time, that indicates a high possibility of such an attack.
	Authentication DoS	1 1 1	ff:d0:bb:25:77:69 ff:ef:45:a2:9b:6d ff:f6:8e:d9:21:41	8c:19:b5:31:6c:e4 8c:19:b5:31:6c:e4 8c:19:b5:31:6c:e4	Possible Authentication DoS attack detected due to high number of Authentication packets coming from multiple MAC addresses. Targeted MACs: 8c:19:b5:31:6c:e4 was targeted 2018 times
	Fake AP beacon flood	*no* *1* *1* ... *1*	*MAC address* *fc:48:41:ad:98:32* *fd:8b:52:61:ee:69* ... *fd:b8:bf:23:f9:7c*	*SSID* *FqeL}CghrlB.goi0* *"NULL"* ... $^{N}D^{H}Hj_{B}DBs\$e3$	Possible Beacon Flood attack detected due to high number of random beacons. If this happens in a high volume and in a small period of time, that indicates a high possibility of such an attack.
	WPS Pin	10	8c:19:b5:31:6c:e4	00:c0:ca:99:42:02	Possible WPS bruteforce attack detected due to high number of EAP packets of WPS with Device Password Authentication Error. Targeted MACs: 8c:19:b5:31:6c:e4 was targeted 10 times - Malicious MACs 00:c0:ca:99:42:02
Host Discovery Network Mapping	ARP scanning	2029	00:0c:29:5e:7c:a7	ff:ff:ff:ff:ff:ff	Possible ARP Scan detected due to high number of ARP packets being transmitted from a single MAC address.Malicious MACs: 00:0c:29:5e:7c:a7 transmitted 2029 packets.
	IP protocol scan	515	192.168.1.39	192.168.1.4	Possible IP Protocol Scan detected due to high number of IPv4 packets being transmitted from a sigle IP address. Target IPs: 192.168.1.4 - Malicious IPs: 192.168.1.39
	ICMP ping sweeps	2 2 ... 2	192.168.1.39 192.168.1.39 ... 192.168.1.39	192.168.232.87 192.168.232.88 ... 192.168.232.99	Possible ICMP Ping sweeping due to high number of ICMP packets being transmitted form a single IP address targeting a subnet. Malicious IPs: 192.168.1.39. If we see a high volume of such traffic destined to many different IP addresses, it means somebody is probably performing ICMP ping sweeping to find alive hosts on the network.
	TCP ping sweeps	2 2 ... 2	192.168.1.39 192.168.1.39 ... 192.168.1.39	192.168.232.87 192.168.232.88 ... 192.168.232.99	Possible TCP Ping sweeping due to high number of TCP packets being transmitted form a single IP address targeting a subnet. Also the packets have window size value 1024 which is very small and unusual and that indicates suspicious traffic.Malicious IPs: 192.168.1.39. If we see a high volume of such traffic destined to many different IP addresses, it means somebody is probably performing TCP ping sweeping to find alive hosts on the network.
	UDP ping sweeps	2 2 ... 2	192.168.1.39 192.168.1.39 ... 192.168.1.39	192.168.232.87 192.168.232.88 ... 192.168.232.99	Possible UDP Ping sweeping due to high number of UDP packets being transmitted form a single IP address targeting a subnet. Also the packets have "Total Length" of 68 which is very small and unusual and that indicates suspicious traffic.Malicious IPs: 192.168.1.39. If we see a high volume of such traffic destined to many different IP addresses, it means somebody is probably performing UDP ping sweeping to find alive hosts on the network.
Network Port Scanning Detection	TCP SYN Scan Stealth Scan	1993	192.168.1.39	192.168.1.4	Possible TCP SYN/Stealth Scan detected. TCP SYN scans probably came from Nmap tool. Target IPs: 192.168.1.4 - Malicious IPs: 192.168.1.39
	TCP Xmass Scan	2000	192.168.1.39	192.168.1.4	Possible TCP Xmass Scan detected. TCP Xmass scans probably came from Nmap tool. Target IPs: 192.168.1.4 - Malicious IPs: 192.168.1.39
	TCP Null Scan	2000	192.168.1.39	192.168.1.4	Possible TCP NULL Scan detected. TCP NULL scans probably came from Nmap tool. Target IPs: 192.168.1.4 - Malicious IPs: 192.168.1.39
	TCP FIN Scan	2000	192.168.1.39	192.168.1.4	Possible TCP FIN Scan detected. TCP FIN scans probably came from Nmap tool. Target IPs: 192.168.1.4 - Malicious IPs: 192.168.1.39
	TCP Connect() Scan	2005 6	192.168.1.39 192.168.1.4	192.168.1.4 142.250.186.78	Possible TCP Connect() Nmap Scan detected. Target IPs: 142.250.186.78, 192.168.1.4 - Malicious IPs: 192.168.1.39. To further investigate this attack, check the TCP Conversation Completeness in Wireshark with the filter "tcp.completeness==39". The number 39 means there were no data transferred in the conversations and no FIN flags set, which is suspicious. If NetSec-Analyzer also displayed a large amount of packets, this indicates a TCP Connect() Nmap scan
	UDP port scan	6	192.168.1.39	192.168.1.4	Possible UDP port Scan detected. UDP port scans probably came from Nmap tool. Target IPs: 192.168.1.4 - Malicious IPs: 192.168.1.39.
Network Attacks Detection	ARP poisoning	7 7	00:0c:29:5e:7c:a7 00:0c:29:5e:7c:a7	14:ab:c5:c4:66:2f 70:97:41:03:0f:40	Possible ARP Poisoning attack detected due to ARP duplicate addresses. Maliciolus MACs: 00:0c:29:5e:7c:a7 - Targeted MACs: 14:ab:c5:c4:66:2f , 70:97:41:03:0f:40
	ICMP flood	18	192.168.1.39	192.168.1.3	Possible ICMP flood attack detected due to multiple ICMP packets transmitted with no data targeting a single IP.Target IPs: 192.168.1.3 Malicious IPs: 192.168.1.39

hackers and security experts frequently use these tools to assess wireless network security and spot possible weaknesses. We also use The Wi-Fi Pineapple Mark VII and the Alfa AWUS036NHA Wi-Fi adapter to conduct denial-of-service attacks.

The results of the use of Wi-Fi-NID testing are presented in Fig. 2. Note that the results shown in Fig. 2 are consistent with the results presented in the early version of this work [14], while the main differences are presented at the end of this section. The results shown correspond to one of the several test networks used during the validation of the tool. In this network, we executed all the attacks outlined in the penetration testing methodology (see Fig. 1). Thus, for each feature of the tool, a part of the results and the final observation (both generated by the tool) are presented.

- DoS attacks feature: The target's MAC address is visible in the event of de-authentication, disassociation, and authentication DoS attacks. In the instance of the fake AP beacon flood, the attack's attempted execution is discovered.

- Host discovery - network mapping feature and network attacks detection feature: These features display the intruder's MAC or IP address in addition to the quantity and size of packets.

- Network port scanning detection feature: Along with other data, this feature also shows the attacker's IP address.

- Brute force WPS pin feature: This feature displays the packets that are harmful together with the targeted MAC and the attacker's device MAC address.

5.1 Initial attack detection validation

For each detection feature supported, every attack underwent 30 individual tests. In all cases the attacks were successfully detected, both the Wi-Fi-specific and the consecutive host discovery and network scanning attacks that followed the successful Wi-Fi attacks and true negatives were not detected.

Some false positives were detected however, in the case of the generic network attacks. For example, in the case of the TCP connect scan attack about 2.2% of the detected packets were eventually false positives. This is due to the fact that the detection of scanning attacks from inside nodes requires setting up very low thresholds, in order to avoid missing actual attacks (false negatives). This however indicates the importance of linking, in Wi-Fi-NID, Wi-Fi-specific attacks as indicators of compromise, for subsequent escalation attacks in the internal network, after a successful penetration of a Wi-Fi network. In this way, the detection of generic network-layer attacks can be fine-tuned, by dynamically increasing or decreasing the detection thresholds, based on events detected at the Wi-Fi layer.

5.2 Wi-Fi attack detection validation

The most important element of this approach is the efficiency, confirmed through rigorous testing under unconventional scenarios where the initial implementation faltered. To simulate these conditions, we developed a script to execute attacks with precision, diverging from the constraints of conventional tools, which increases the complexity of the test scenarios. We encountered several factors in our diverse tests that underscored the need for a more efficient method, which we will discuss next. It's worth noting that by using well-known, conventional tools that conduct de-authentication attacks, such as aireplay-ng or mdk3 and mdk4, they could be easily detected as they send a significant volume of de-authentication packets per second, approximately 100-400 packets per second. Hence, the outliers in the network traffic could be easily identified without false positives.

In our initial implementation [14], the use of a static threshold for attack detection posed considerable issues, including frequent false positives, prompting us to consider a dynamic threshold. This adjustment significantly minimized false positives, as described in Section 3.1.

The primary difference between the two versions of our framework lies in the implementation of static versus dynamic thresholds. To evaluate the efficiency difference, we replicated the same scenario in both versions. Specifically, we chose to implement the de-authentication attack using available custom attack tools and our custom attack tool, each with varying parameters.
In more detail, 100 tests were conducted on each framework. Although both frameworks (the initial version of [14] and the enhanced dynamic framework presented in this paper) showed a high accuracy (exceeding 90% even in the initial version), the framework with the static threshold failed to detect any attack, whether sophisticated or otherwise, with a packet count below the static threshold. In contrast, the framework with the dynamic threshold succeeded in detecting all instances except some corner cases (such as setting the time interval set to more than six seconds). However, such an attack would not be successful in practice.
Additionally, we enhanced our approach to accurately identify the attacker's MAC address, a detail overlooked in the initial methodology that occasionally misidentified the attacker's access point instead of the device's MAC address. Bellow is the message in such cases:
Attack Detected Main targeted MAC Address: 70:97:41:03:0f:49 (targeted 16364 times) Main targeted MAC Address associated with 70:97:41:03:0f:49: 0a:e8:83:1a:52:6e (targeted 16364 times)

For instance, we tested the system's resilience by launching de-authentication attacks with minimal packets to avoid detection. As shown in Fig. 5, our refined methodology proved robust, successfully identifying the attack irrespective of the packet count or the timing disparities in the network's packet transactions. This outcome underscores the enhanced adaptability and accuracy of our implementation in identifying threats.

To explore this scenario and test our tool, we created a custom attack tool to carry out stealth de-authentication attacks. This tool provides us with the ability to construct de-authentication packets from scratch and reduce the number of packets sent per second, as well as the frequency at which they are sent, blending them with normal traffic. This makes the attack harder to detect. In this way, we investigated scenarios where an attacker tries to evade detection, which allowed us to test the effectiveness of our tool in such scenarios.

As mentioned earlier, non-outliers are assigned a value of 0, while outliers retain their original values. In our approach to detecting anomalies in a scenario like the one described above, we examine each data point in relation to its neighboring data points. If a data point is surrounded by at least five consecutive zeros (meaning it has a sequence of zeros both before and after it), it is not considered a candidate for an attack point in malicious traffic. This is because it indicates that, in the five seconds preceding and following the spike in de-authentication packets, there were no other spikes exceeding the threshold value. Therefore, it lacks the characteristics of a successful de-authentication attack as the continuity of the attack breaks and considering our research we concluded that usually in this time period a system would have enough time to reconnect again to the access point before

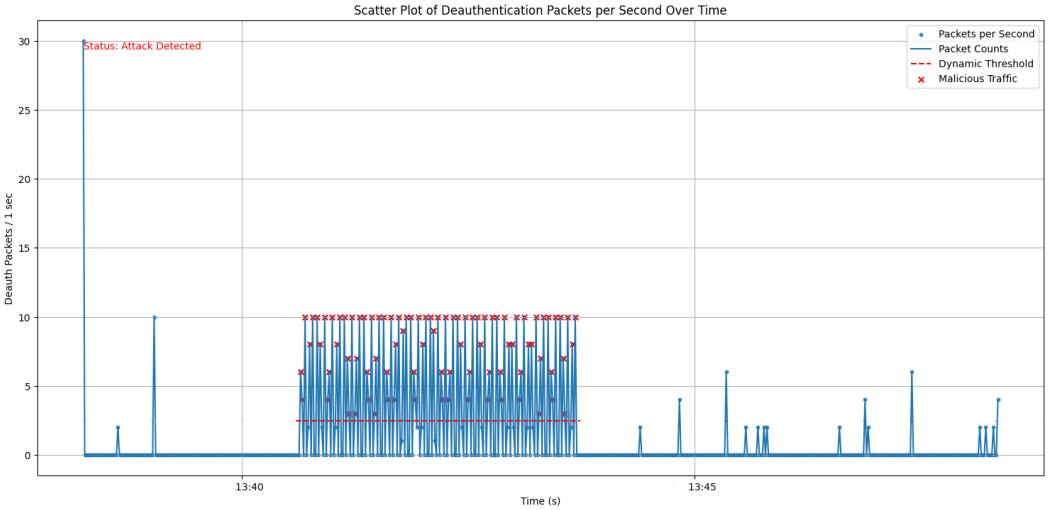


Fig. 5 – Custom de-authentication attack packet transactions

it gets de-authenticated again. However, if any outlier is identified with any neighboring data points within a 5-second window that surpasses the dynamically calculated threshold, it is considered an attack point and marked as malicious traffic.

Therefore, in comparison to our initial version [14] Wi-Fi-NID has now advanced to a point where it can accurately identify stealth attacks, including those involving only a handful of packets that attempt to slip past security measures unnoticed. It was shown to be capable of distinguishing these suspect activities from regular network traffic, eliminating confusion from false alerts that can occur when the router transmits more packets than what the security system recognizes as typical.

For example, as we can see in Fig. 5, our tool was able to detect this successful attack even when the packets sent per second were minimal (3-10 packets) compared to those in the previous image (90-230 packets). Furthermore, it managed to distinguish between malicious packets and legitimate ones, as there are spikes where, at that moment, the router sent a considerable number of de-authentication packets to the client, well exceeding the dynamic threshold, as well as the packets from the attacker per second.

6. CONCLUSION

The pen-testing methodology for Wi-Fi networks that we presented in this paper is based on the techniques and resources that attackers use to compromise these networks. We implemented the methodology in Wi-Fi-NID, a software tool that eavesdrops on Wi-Fi network traffic and employs a layered analysis to detect malicious activity. Wi-Fi-NID detects attack patterns initiated at the Wi-Fi layer by employing dynamic detection thresholds, in order to detect Wi-Fi intrusion attempts and to concurrently detect and avoid outliers, by applying statistical analysis based on the rolling median. This was based on the observation that Wi-Fi-specific attacks such as de-authentication and disassociation packets aren't common, but show up more frequently when such attacks are happening. Then, traces and important details about the attacker, such as suspected MAC and IP addresses, are further investigated in the second phase of the methodology, in order to detect additional attack patterns that indicate lateral movements at the internal network. Wi-Fi-NID attempts to apprehend the attacker during actions that are hard to evade. Our current proof of concept is deployed in a test environment across multiple real-world Wi-Fi networks.

A pivotal aspect of our strategy is the future adaptability of our system into a Machine Learning (ML) framework. The custom algorithms and mathematical models we developed are designed with an inherent flexibility that will allow them to be integrated into a more complex ML model in the future. As a future step we will focus on enhancing the tool with additional statistic measures, towards a more sophisticated self-learning defense mechanism. This transition has been partially realized, by implementing dynamic detection thresholds based on rolling median. We aim to elaborate on these advancements in subsequent work. Furthermore, we will try to create and then train this dataset so that we can detect attacks using pure AI methods. It is also an opportunity to see how much more efficient the AI methods are compared to the older ones. More specifically, the first step is to create our own dataset for de-authentication attacks, in order to use it to test algorithms such as Long Short-Term Memory (LSTM) and Isolation forest. In addition, we plan to extend the tool to an active prevention tool that beyond detection will immediately mitigate malicious attempts to penetrate an 802.11 network in real time.

REFERENCES

[1] *Network Forensics Toolset, Document for students*. 2015. URL: https://www.enisa.europa.eu/topics/training-and-exercises/trainings-for-cybersecurity-specialists/online-training-material/documents/network-forensics-toolset.

[2] Department of Homeland Security. "FY 2019-2020 Annual Performance Report". In: (2020), p. 8. URL: https://www.dhs.gov/sites/default/files/publications/dhs_fy_2019-2021_apr_final.pdf.

[3] Rami Ahmad, Raniyah Wazirali, and Tarik Abu-Ain. "Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues". In: *Sensors* 22.13 (2022). ISSN: 1424-8220. DOI: 10.3390/s22134730.

[4] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Stefanos Gritzalis. "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset". In: *IEEE Communications Surveys and Tutorials* 18.1 (2016), pp. 184–208. DOI: 10.1109/COMST.2015.2402161.

[5] K A Scarfone and P M Mell. *Guide to Intrusion Detection and Prevention Systems (IDPS)*. Tech. rep. NIST SP 800-94. National Institute of Standards and Technology, 2007. DOI: 10.6028/NIST.SP.800-94. (Visited on 03/22/2023).

[6] Neil Dalal, Nadeem Akhtar, Anubhav Gupta, Nikhil Karamchandani, Gaurav S. Kasbekar, and Jatin Parekh. "A Wireless Intrusion Detection System for 802.11 WPA3 Networks". In: *2022 14th International Conference on COMmunication Systems and NETworkS (COMSNETS)*. 2022, pp. 384–392.

[7] Tianjun Chen, Dexin Kong, and Yuxin Hong. "Development and Implementation of Anti Phishing Wi-Fi and Information Security Protection APP based on Android". In: *IOP Conference Series: Earth and Environmental Science*. Vol. 1802. 3. IOP Publishing Ltd, 2021. DOI: 10.1088/1742-6596/1802/3/032109.

[8] Gustavo A. Nunez Segura, Sotiris Skaperas, Arsenia Chorti, Lefteris Mamatas, and Cintia Borges Margi. "Denial of Service Attacks Detection in Software-Defined Wireless Sensor Networks". In: *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2020, pp. 1–7. DOI: 10.1109/ICCWorkshops49005.2020.9145136.

[9] Bhagyashri Tushir, Yogesh Dalal, Behnam Dezfouli, and Yuhong Liu. "A Quantitative Study of DDoS and E-DDoS Attacks on WiFi Smart Home Devices". In: *IEEE Internet of Things Journal* 8.8 (2021), pp. 6282–6292. DOI: 10.1109/JIOT.2020.3026023.

[10] Pragati Shrivastava, Mohd Saalim Jamal, and Kotaro Kataoka. "EvilScout: Detection and Mitigation of Evil Twin Attack in SDN Enabled WiFi". In: *IEEE Transactions on Network and Service Management* 17.1 (2020), pp. 89–102. DOI: 10.1109/TNSM.2020.2972774.

[11] George Chatzisofroniou and Panayiotis Kotzanikolaou. "Exploiting WiFi usability features for association attacks in IEEE 802.11:: Attack analysis and mitigation controls". In: *Journal of Computer Security* 30.3 (2022), pp. 357–380.

[12] Ayong Ye, Qing Li, Qiang Zhang, and Baorong Cheng. "Detection of Spoofing Attacks in WLAN-Based Positioning Systems Using WiFi Hotspot Tags". In: *IEEE Access* 8 (2020), pp. 39768–39780. DOI: 10.1109/ACCESS.2020.2976189.

[13] İlhan Fırat KILINÇER, Fatih ERTAM, and Abdülkadir ŞENGÜR. "Automated Fake Access Point Attack Detection and Prevention System with IoT Devices". In: *Balkan Journal of Electrical and Computer Engineering* 8.1 (2020), pp. 50–56. DOI: 10.17694/bajece.634104.

[14] Dimitris Koutras, Panos Dimitrellos, Panayiotis Kotzanikolaou, and Christos Douligeris. "Automated WiFi Incident Detection Attack Tool on 802.11 Networks". In: *2023 IEEE Symposium on Computers and Communications (ISCC)*. 2023, pp. 464–469. DOI: 10.1109/ISCC58397.2023.10218077.

[15] Joseph R Rose, Matthew Swann, Gueltoum Bendiab, Stavros Shiaeles, and Nicholas Kolokotronis. "Intrusion Detection using Network Traffic Profiling and Machine Learning for IoT". In: *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*. 2021, pp. 409–415. DOI: 10.1109/NetSoft51509.2021.9492685.

[16] M. Nivaashini and P. Thangaraj. "Computational intelligence techniques for automatic detection of Wi-Fi attacks in wireless IoT networks". In: *Wireless Networks* 27.4 (May 2021), pp. 2761–2784. ISSN: 1572-8196. DOI: 10.1007/s11276-021-02594-2.

[17] Neil Dalal, Nadeem Akhtar, Anubhav Gupta, Nikhil Karamchandani, Gaurav S. Kasbekar, and Jatin Parekh. "A Wireless Intrusion Detection System for 802.11 WPA3 Networks". In: *2022 14th International Conference on COMmunication Systems and NETworkS (COMSNETS)*. 2022, pp. 384–392. DOI: 10.1109/COMSNETS53615.2022.9668542.

[18] George Chatzisofroniou and Panayiotis Kotzanikolaou. "Exploiting WiFi usability features for association attacks in IEEE 802.11: Attack analysis and mitigation controls". In: *Journal of Computer Security* 30 (2022). 3, pp. 357–380. ISSN: 1875-8924. DOI: 10.3233/JCS-210036.

[19] Irfan Yaqoob, Syed Adil Hussain, Saqib Mamoon, Nouman Naseer, Jazeb Akram, and Anees Rehman. "Penetration testing and vulnerability assessment". In: *Journal of Network Communications and Emerging Technologies (JNCET)* 7.8 (2017).

[20] Eric Filiol, Francesco Mercaldo, and Antonella Santone. "A Method for Automatic Penetration Testing and Mitigation: A Red Hat Approach". In: *Procedia Computer Science* 192 (2021). Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 25th International Conference KES2021, pp. 2039–2046. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.08.210.

[21] Fairuz Zahirah Lidanta, Ahmad Almaarif, and Avon Budiyono. "Vulnerability Analysis of Wireless LAN Networks Using Penetration Testing Execution Standard: A Case Study of Cafes in Palembang". In: *2021 International Conference on ICT for Smart Society (ICISS)*. Aug. 2021, pp. 1–5. DOI: 10.1109/ICISS53185.2021.9533216.

[22] Matthew Denis, Carlos Zena, and Thaier Hayajneh. "Penetration testing: Concepts, attack methods, and defense strategies". In: *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. Apr. 2016, pp. 1–6. DOI: 10.1109/LISAT.2016.7494156.

[23] Thomas Girdler and Vassilios G. Vassilakis. "Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses". In: *Computers and Electrical Engineering* 90 (2021), p. 106990. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2021.106990.

[24] Mayoon Yaibuates and Roungsan Chaisricharoen. "A Combination of ICMP and ARP for DHCP Malicious Attack Identification". In: *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT and NCON)*. 2020, pp. 15–19. DOI: 10.1109/ECTIDAMTNCON48261.2020.9090760.

[25] Fu-Hau Hsu, Yu-Liang Hsu, and Chuan-Sheng Wang. "A solution to detect the existence of a malicious rogue AP". In: *Computer Communications* 142-143 (2019), pp. 62–68. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2019.03.013.

[26] M. Anathi and K. Vijayakumar. "An intelligent approach for dynamic network traffic restriction using MAC address verification". In: *Computer Communications* 154 (2020), pp. 559–564. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2020.02.021.

[27] Jian Wang, Nicolas Juarez, Emma Kohm, Yongxin Liu, Jiawei Yuan, and Houbing Song. "Integration of SDR and UAS for Malicious Wi-Fi Hotspots Detection". In: *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. 2019, pp. 1–8. DOI: 10.1109/ICNSURV.2019.8735296.

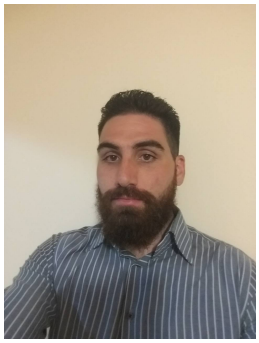
[28] G. Pradeepini, G. Muni Sai, and V. Aruna. "Hybrid Pcap analyser using T-Shark a tool that makes use of open source analyser that can Meet Industrial Standards". In: *International Journal of Engineering and Technology(UAE)* 7.4 (2018), pp. 85–88. ISSN: 2227524X. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85055619963&partnerID=40&md5=48a315fcb5609ab7244d408bc93851e7.

[29] Luca Deri, Maurizio Martinelli, Tomasz Bujlow, and Alfredo Cardigliano. "nDPI: Open-source high-speed deep packet inspection". In: *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2014, pp. 617–622. DOI: 10.1109/IWCMC.2014.6906427.

[30] Wenguang Song, Mykola Beshley, Krzysztof Przystupa, Halyna Beshley, Orest Kochan, Andrii Pryslupskyi, Daniel Pieniak, and Jun Su. "A Software Deep Packet Inspection System for Network Traffic Analysis and Anomaly Detection". In: *Sensors* 20.6 (2020). ISSN: 1424-8220. DOI: 10.3390/s20061637.

[31] Mamoru Mimura. "Adjusting lexical features of actual proxy logs for intrusion detection". In: *Journal of Information Security and Applications* 50 (2020), p. 102408. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2019.102408.

AUTHORS



Dimitris Koutras is a PhD candidate in network security, at the University of Piraeus, Department of Informatics. He obtained a bachelor's degree in Information technology and computer technology engineering from the Technological educational institute of Central Greece (Lamia-2016). Then he also obtained a Master of Science (M.Sc.) in security management engineering from the University of Piraeus (Piraeus-2019). He currently participates in European research programs as a researcher of University of Piraeus Research center. He also participates as a trainer in various cybersecurity seminars, concerning maritime, healthcare and supply chain. He holds an ISO 27001:2013 lead auditor, certification in information security. His research work concerns network security analysis, risk assessment and operating systems security. Dimitris has collaborated

with UPRC on several major projects, including CyberSec for Europe, CyberSecPro, MELITY and ARTEMIS, funded by the European Union and the Greek government. Dimitris is also a published author, with five publications, one of which was praised as an editor's choice.

Panos Dimitrellos born in 1999, is a computer science graduate from the University of Piraeus, with a bachelor of science degree earned in 2022. Following the completion of his studies, he joined the Greek army, where he held the role of a cybersecurity engineer and a SOC analyst at the Hellenic National Defence General Staff's Directorate of Cyber Defence in Athens until 2023. In parallel to his military service, Panos dedicated himself to research projects based in cybersecurity conducted in collaboration with the Research Center of the University of Piraeus. These ongoing research initiatives delve into the intricate realms of network security analysis, penetration testing, and cybersecurity engineering.

Panayiotis Kotzanikolaou is an associate professor in network security and privacy at the University of Piraeus, Department of Informatics and the director of the MSc in cybersecurity and data science. He has a degree in computer science (1998) from the University of Piraeus and a Ph.D in ICT security (2003). Formerly, has served as a security auditor at the Hellenic Authority for the Security and Privacy in Communications (ADAE), and has also worked as a security consultant in the private sector. He has participated in various national and European R&D projects. He has participated as a Program Committee member in international conferences and he is a reviewer in various international journals. He has published more than 100 papers in books, journals and international conferences. He is a member of the Greek Computing Society and has received various certifications in information security (CISSP, ISO 27001 Lead Auditor).

Christos Douligeris is a professor at the Department of Informatics, University of Piraeus, Greece. Formerly, he held positions with the Department of Electrical and Computer Engineering at the University of Miami. He was an associate member of the Hellenic Authority for Information and Communication Assurance and Privacy and the President and CEO of Hellenic Electronic Governance for Social Security SA. Dr. Douligeris has published extensively in the networking scientific literature and he has participated in many research and development projects. He is the co-editor of a book on ''Network Security'' published by IEEE Press/ John Wiley and he is on the editorial boards of several scientific journals, as well as on the technical program committees of major international conferences. He has been involved extensively in curriculum development both in the USA and Greece. His latest work has focused on the use of big data and artificial intelligence techniques in several areas, mainly in telecommunications planning and management and in security analysis of port information systems. Moreover, he has been working in data analytics techniques in learning and education and emergency response oerations. Prof, Douligeris is the director of the Network Research Lab (http://netlab.cs.unipi.gr/gr/), which is closely cooperating with the Security Lab.