Potential analysis for the detection of attacks on wireless networks using the Wireless Intrusion Detection System Nzyme

Eisenhut, Maximilian[1], Honekamp, Wilfried[2]

Abstract: Due to the flexibility and low cost of acquisition compared to wired network connections, wireless networks continue to proliferate. Due to this increasing number and the characteristics of a shared medium, it offers potential attackers a suitable platform to easily gain access to diverse network types. To this end, the range of specialised hardware and software for attacking wireless networks is constantly evolving. Information on the location and other parameters of wireless networks is also documented and updated online in a largely automated manner. Particularly in the economic as well as in the public environment, a special need can thus arise to detect attacks, identify attackers and initiate countermeasures on the basis of this information. This paper describes the evaluation of the possibilities offered by the open-source Wireless Intrusion Detection System (WIDS) Nzyme. For this purpose, the messages that occur during different attacks were examined. Furthermore, real data was recorded and evaluated based on the parameters from the test attacks to draw conclusions about the type and frequency of attacks. The ratio between legitimate reports and false alarms was also determined. Test attacks were successfully detected and could be assigned to possible attacks. Real data was recorded at three locations and compared with the patterns from the test attacks. The evaluation shows that the rate of false alarms in real operations is unacceptable, at over 27%. The causes for this are mostly misconfigurations and atmospheric disturbances. The study further shows, that combined alarm messages allow conclusions to be drawn about the type of attack carried out and thus the number of false alarms can be reduced. The effort and benefit of a WIDS are currently not yet in a meaningful relationship. Nevertheless, use and further development are recommended, taking these circumstances into account.

Keywords: Anomaly detection, intrusion detection system, Nzyme, WLAN.

1 Introduction and Background

Due to the advantages of wireless local area networks (WLAN) in terms of flexibility and cost-effectiveness compared to wired networks, they are becoming more and more widespread, whether as public access points (AP) to the Internet, for automation in the business environment, or for use by public authorities [Wi22a]. Bandwidth and speed are increasing due to the further development of technologies. The aspect of security is also

[1] Hochschule Stralsund, Fakultät für Elektrotechnik und Informatik, Zur Schwedenschanze 15, Stralsund, 18435, maximilian.eisenhut@hochschule-stralsund.de

[2] Hochschule Stralsund, IACS, Zur Schwedenschanze 15, Stralsund, 18435, wilfried.honekamp@hochschule-stralsund.de, https://orcid.org/0000-0003-2931-7047

doi:10.18420/inf2022_03

subject to constant development through the implementation of new security standards and protocols for the encrypted transmission of data.

On the other hand, there is the growing complexity of the IT infrastructure and the associated potential for vulnerabilities in these systems. In addition, there is the increasing public availability of information on the nature and geographical locations of wireless networks worldwide [Wi22b]. The increasing supply of hacking hardware and software also favours the abusive use of this technology. Two fundamental interests collide in this area: on the one hand, the goal of finding vulnerabilities in systems through observation and attacks and exploiting them for set goals. And on the other hand, to be able to take appropriate countermeasures to protect the basic goals of information security, confidentiality, integrity and availability. Thus, there is a constant interplay between attack and defence. "WiFi will continue to be more vulnerable to attack than hardwired LAN as long as electromagnetic radiation fails to obey property lines." [Be05], (p. 28)

One possible defence measure here can be the use of a system that checks network traffic for anomalies and stores evidence data. This is important for a forensic investigation during or after an incident. Especially for wireless networks, due to their physical nature, there may be a relevance to use such so-called wireless intrusion detection systems (WIDS). Different systems are offered on the market, e. g. Graylog, Nzyme, or Wirebug. The possibilities of data evaluation from intrusion detection systems (IDS), for this purpose, are described, among others, in the article by Kasongo and Sun "A Deep learning method with filtered based feature engineering for wireless intrusion detection system" [Ka19]. The use of an explicit data set, for machine learning based on data mining, is not intended to be the content of this work. Instead, this work addresses the current relevance of WIDS and discusses whether a dedicated open source WIDS can contribute to the security of wireless networks.

An IDS is a device or software that attempts to detect network attacks in order to prevent possible incidents/attacks by collecting and analysing data. To ensure security, IDS have already been developed for use in the WLAN, known as WIDS. Similar to traditional IDS, these WIDS can detect patterns of known attacks, identify anomalous network activity, and detect WLAN policy violations by monitoring and analysing network, user, and system activity. Like traditional signature-based IDS and anomaly-based IDS, WIDS can detect anomalies according to either pre-defined signatures (known threats) or observed abnormal (exceeding baselines) network behaviour. WIDS can be either centralised or decentralised. In a centralised WIDS, the central management system combines and analyses all wireless data from each distributed individual sensor. In a decentralised WIDS, there is more than one device that both collects data and generates the intrusion alerts by analysing the data. [Ya13]

2 Method

The suitability of an open source WIDS in practice is to be examined by means of a case study using Nzyme as an example. It is a decentralised open-source solution, and thus meets the requirements to be operated independently at different measurement sites. The system is designed as an explicit wireless intrusion detection system. The software runs on Raspberry Pi OS or Ubuntu Server, is implemented in Java and uses a PostgreSQL database.

The relationship between legitimate alerting and the output of false alarms will also be examined. The differentiation of the attacks is done by comparing the recorded data with alarm patterns detected during test attacks. In order to check whether a conclusion can be drawn about the executed attack based on differently occurring alarm messages or combinations of these, the attack categories spoofing, denial of service and deauthentication were recorded as manual attacks. Furthermore, the attacks were carried out with dedicated hacking hardware in order to differentiate them from automated attacks. In addition, a Kr00k attack was executed on a matching client device to further differentiate the detection. The Kr00ker attack is a variant of the "Krack" attack, which exploits a vulnerability in the 4-way handshake for authentication between the client device and the base station. Both devices use a pairwise master key (PMK). In this case, the signal is delayed by an attacker so that the ascending packet number between the client device and the server differs greatly, causing the connection to be reset. In the process, the PMK is set to 0. This allows the attacker to read the entire traffic without encryption.

To represent different applications and environments of WLAN, monitoring was done at three locations (university, local government, private residence) and the alarm messages that occurred were recorded. After installing the monitoring stations, a pilot study was conducted over two days for configuration. During this time, the network traffic occurring at the site, the legitimate fingerprints, channels and basic service set identifier (BSSID) of the AP, and the thresholds of the de-authentication frames were recorded. The average power level of the monitored AP was recorded to provide a reference value for detecting deviations, such as those caused by rogue AP. Based on this, false alarms can be minimised and it can be determined which attack messages or combinations, occur in which attack. The minimisation of false alarms plays an important role, since too frequently occurring false alarms, lead to a decrease of the sensitivity of the observing place, and so with a genuine alarming, by the observing person, no action takes place. [Gr66]

To perform the test attacks, airgeddon [V117] was run as a bash script on a PC running Kali Linux [Of22]. This is freely accessible, largely automated, and covers multiple attack patterns, such as current denial of service, de-authentication, offline decrypt, evil-Twin, and enterprise attacks. The Kr00ker attack was executed based on a proof-of-concept script. Attacks using a copied specific BSSID were carried out manually using air-base-ng and macchanger. To complement this, a fake AP was set up [Ha14].

The AP to be monitored can be stored with fingerprint, channel, BSSID and SSID. A threshold value for the number of de-authentication frames to trigger an alarm message can be defined. The address and access data of the system can be stored for alerting via e-mail. The address and access data for an existing logging system can be stored in the configuration for the central recording of data. This is used to be able to carry out general evaluations independent of location. To record the data, 3 measuring stations were set up. Two based on a Raspberry Pi 4 and one based on a Raspberry Pi 3B. A tracker for the localisation of possible rogue AP was set up on the basis of a Raspberry Pi 4 and connected via compatible long range (LoRa) modules over LoRa WAN with the measuring station ECTO-1 (Figure 1) located at the university.

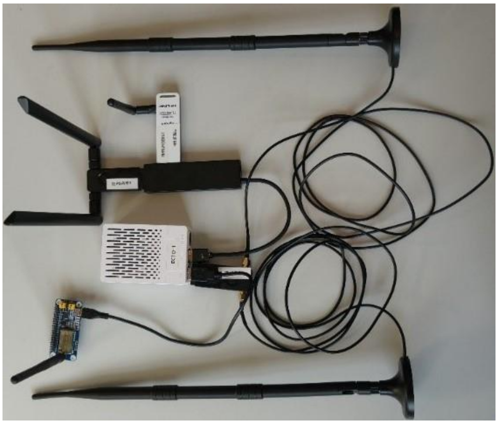


Figure 1: Nzyme receiving station ECTO-1

As a wireless network adapter, a USB stick with RT5572 chipset and two antenna connections, was operated in monitor mode to monitor the defined networks in the 2.4 GHz and 5GHz bands. In monitor mode, no connection is established to a specific AP, but all data packets within range are picked up. To increase the range, the device was connected to two 9dBi antennas. The network adapter with two antennas simulates as "beacon_trap" an AP with SSIDs without underlying infrastructure to provoke attacks. If this network is offered by a device, it is an attack. Another USB device with an antenna simulates a cell phone that is to be used to connect to known networks. For this purpose, fictitious SSID are requested by the device. As soon as these are offered by a rogue AP within range of the WIDS, the "probe_response_trap" is triggered and issues a corresponding alarm message. Alarms provoked in this way can be localised via the fingerprint of the rogue AP using a tracker. The tracker is connected to the measuring station via LoRa Wan. The background for this is the low susceptibility of LoRa WAN to jamming attacks on WLAN. [Ös18]

As an example, a tracker based on a Raspberry Pi 4 with RT5572 USB WLAN adapter and LoRA WAN module, in combination with a tablet as display, was set up for testing purposes. The localisation of suspicious devices, such as rogue AP, is implemented on the basis of a measurement of the recorded power level. The measurement is in decibel milliwatts (dBm); the lower the negative value, the stronger the signal or the shorter the distance to the transmitter [Ri21]. Two further measuring stations were implemented as ECTO-2 in a domestic environment and ECTO-3 in an official environment. These were operated without "trap" and tracker, as a pure base station, in order to further differentiate the accruing data situation. Incoming messages were recorded on a logging server using the open-source software Graylog. All incoming messages were recorded. For data protection reasons, the data on other wireless networks is not evaluated. In production operation, MAC addresses of client devices and new AP would be recorded. The recorded messages are sorted by timestamps. Timestamps without messages are discarded. Subsequently, implausible messages are removed. Implausible messages are alarm messages for which only one message occurred within half an hour or which are contradictory.

For the evaluation of the alarm messages, they are filtered in the first step and separated according to the respective measuring point. Entries without message or single entries, except with the message "deauth_flood", are removed. The sum of the individual messages by type and number is recorded. Entries with multiple simultaneous messages are grouped and summed. Entries with an interval of less than thirty minutes are grouped and summed. In order to draw conclusions about the attack carried out, the entries are compared with the recorded patterns of the test attacks. Based on this, the number of false alarms and the type, duration and frequency of authentic alarm messages can be determined. This comparison serves as a basis for making a statement about the sensible use of the WIDS Nzyme.

3 Results

When the attacks started, the data was recorded and the corresponding alarm patterns were assigned. In the time period from 1/2/2022 to 3/6/2022, a total of 79,897,248 frames were recorded using the Graylog software [Gr19]. The majority of these, 79,875,714, were beacon frames from AP within range of the WIDS. They do not contain any anomalies.

Table 1: Recorded spoofing patterns

	beacon_rate_anomaly	crypto_change	multiple_signal_tracks	unexpected_bssid	unexpected_channel	unexpected_fingerprint

Spoofing: other encryption, other channel, same SSID, other BSSID		x		x	x	
Spoofing SSID: different encryption, same channel, same SSID, different BSSID		x		x		
Spoofing SSID: different encryption, same channel, same SSID, same BSSID	x	x	x			x
Spoofing SSID: same encryption, same channel, same SSID, same BSSID*.	x		x			x
Client device spoofing*						

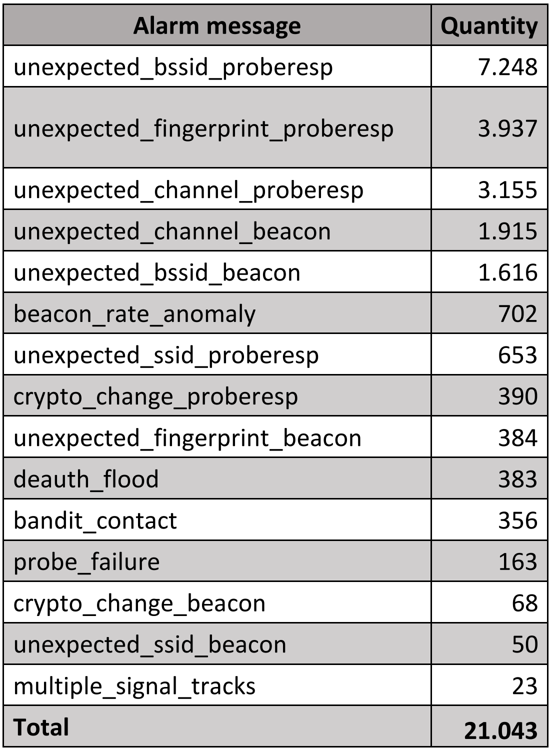
*manually executed attacks

The beacon frames contain, among other things, parameters on SSID, signal strength, channel and supported encryption at the time of recording. The number of recorded alarm messages is 21,534, i. e. 0.027% of the total number of recorded messages. The alarm message "unknown_ssid" occurred 491 times. This message occurs when an SSID is detected for the first time in the reception range of the WIDS. In the process, the beacon frames of WLAN that are not monitored by WIDS are captured, which is of no further relevance for this work leading to a remaining no. of 21,043 (approx. 31% of ECTO-1, 2% of ECTO-2, and 67% of ECTO-3) alarms that can be seen in Table 2. The use of the alarm message in conjunction with other messages to detect an attack is also not relevant, as other alarm messages such as "unexpected_fingerprint_beacon" (used fingerprint is not in the expectation list) occur. It should also be added that beacon frames allow conclusions to be drawn about personal data. Thus, the protection of personal data is another aspect to reject these alarm messages. Table 1 describes the alarm messages that occurred simultaneously depending on the attack that was carried out. The rows contain the respective attack and the columns the alarm message that occurred.

The results in Table 1 show that different spoofing attacks generate different patterns. Thus, the alarm message "crypto_change_beacon" or "crypto_change_response", in combination with other occurring alarms, can be assumed as a sure sign for a rogue AP in range. For example, a spoofing attack with different encryption, different channel, same SSID, different BSSID simultaneously generated the "beacon_rate_anomaly" (beacon frames are sent in a higher frequency than expected), "crypto_change", "unexpected_bssid" and "unexpected_channel" messages. A spoofing attack with different encryption, same channel, same SSID, different BSSID, generates the messages "crypto_change" and "unexpected_bssid". The spoofing attack with different encryption, same channel, same SSID, same BSSID, generates the messages: "beacon_rate_anomaly", "crypto_change", "multiple_signal_tracks" (more than one signal track is transmitted), "unexpec-ted_fingerprint". The spoofing attack with same encryption, same channel, same SSID, same BSSID generates the messages: "beacon_rate_anomaly", "mul-tiple_signal_tracks", and "unexpected_fingerprint". Client device spoofing with the same encryption and Mac address of an end device connected to the AP did not generate any message in the WIDS, but caused disconnections that can be noticed by the user. The denial-of-service attacks generated the message "deauth_flood" over the duration of the attack. In contrast, the de-authentication attack to record handshakes only recorded the "deauth_flood" message selectively in a timestamp.

For attacks with dedicated devices, such as Pineapple and Pwnagotchi, the messages "bandit_contact" were recorded in addition to the patterns described above. During the deauthentication attack by a Pwnagotchi, the message "pwnagotchi_advertisement" was additionally recorded. When the Krack and Kr00k attacks were launched, a "deauth_flood" message was recorded during the test attacks. No other messages could be recorded because no device with an appropriately vulnerable chip was available. Fifteen different individual types of alarms were recorded. As the test attacks show, the messages can be divided into secure and insecure types. Secure types are alarm messages that can occur individually and clearly signal an attack in range. An example of this in this work is "bandit_contact". Bandit describes hardware that is explicitly used for spying, jamming and manipulating.

Table 2: Number of alarm messages recorded



Alarm message	Quantity
unexpected_bssid_proberesp	7.248
unexpected_fingerprint_proberesp	3.937
unexpected_channel_proberesp	3.155
unexpected_channel_beacon	1.915
unexpected_bssid_beacon	1.616
beacon_rate_anomaly	702
unexpected_ssid_proberesp	653
crypto_change_proberesp	390
unexpected_fingerprint_beacon	384
deauth_flood	383
bandit_contact	356
probe_failure	163
crypto_change_beacon	68
unexpected_ssid_beacon	50
multiple_signal_tracks	23
Total	**21.043**

In the following, alarm messages that cannot be clearly assigned to an attacker or that only define an alarm in combination with other messages occurring at the same time are defined as unsafe. Table 2 shows the distribution of alarms recorded without further checking for possible duplicate or false alarms. In order to be able to make well-founded statements about alarms that have occurred, it is therefore necessary to further investigate and clean the data to exclude false-positive messages. The basis for this is the knowledge gained from the test attacks. Furthermore, it is not clear which alarm messages were recorded at which location and whether they can be correlated with each other for attack detection. In order to draw conclusions about possible attacks and to be able to classify the data according to relevance, it makes sense to correlate the data according to location and time of recording. It also makes sense to look at individual alarms in order to identify attacks that can be clearly classified as attacks or false alarms.

Alarm messages, considered individually, can be divided into the following two categories:

- Authentic alerts, which clearly mark an attack or can be identified as false alerts depending on the alert message when they occur individually.
- Messages that can only indicate an attack in conjunction with messages that occur at the same time (based on the findings from the test attacks).

Authentic alarms are "bandit_contact", "multiple_signal_track", "unexpected_fingerprint" and "unexpected_bssid". The messages "crypto_change", "unexpected_ssid", "unexpected_channel" and "beacon_rate_anomaly" in combination with other messages indicate an attack. The message "deauth_flood" must be considered separately, as it can indicate a deauthentication attack to record a 4-way handshake or, in conjunction with "unexpec-ted_fingerprint", an "Evil Twin".

4 Discussion and Conclusions

From the recorded data and the evaluation, it can be seen that a WIDS offers a possibility to significantly increase the security of WLAN. In summary, the messages "bandit_contact", "multip-le_signal_track", "unexpected_fingerprint" and "crypto_change", and thus four of the nine messages considered, can be regarded as a sure sign of an attack. The other messages cannot be considered safe on their own. These are the messages "deauth_flood", "unexpected_channel", "unexpec-ted_bssid", "unexpected_ssid" and "beacon_rate_anomaly". This increases the risk of false alarms. Careful configuration of the WIDS can minimise the risk, but not completely eliminate it. The configuration requires knowledge of the structure and operation of a WLAN, possible attacks on it and the detection of these attacks.

With this knowledge, the combination of individual alarms into patterns can further reduce the risk of false alarms. In addition, these combinations can be considered as a basis for machine learning or the use of other technologies, such as a Bayesian network. Basically, the rate of false alarms of approximately 28% is unacceptable for the use of the tested system. With an immense number of false alarms, the attention span of humans is too short, loosely based on the fable "The Shepherd Boy and the Wolf". In addition, the time required to manually identify an alarm as an authentic message or as a false alarm is too long. This is not in proportion to the number of confirmed alarms, so this process must be automated. Machine learning helps to reduce the rate of false alarms and improve the quality and speed of detection of attacks. By combining this technology with Nzyme, the advantages of the high detection rate can be combined with the possibilities offered by Nzyme. In this way, a device can be reliably identified as an attacker and tracked down with a tracker. In order to be able to make a reproducible statement about detected attacks versus attacks that have actually taken place, a separate observation under laboratory conditions is necessary.

The use of AP to determine the position can be manipulated and thus people can be misled. A WIDS can help to detect these attacks. Attacks on WLAN in industrial and public areas can have an impact on the control networks in public and industrial companies. The use of a WIDS helps to detect these attacks and to avoid disturbances. The distribution of alarm messages over days of the week may indicate that attacks are carried out by persons working with the system. The questions of motivation and background, as well as frequency and distribution, are the subject of divergent work. The question also remains open whether and how successful key reinstallation attacks such as Krack or Kr00K are detected by wireless intrusion detection systems, which can be the subject of future work. The possibilities of evaluating commercial WIDS also offer potential for further work. In summary, the use of the WIDS Nzyme only has the potential to contribute to security when the number of false alarms is significantly reduced.

The findings of this work can form the basis for future work. For example, a comparison of different WIDS would be desirable, or the analysis of attacks on WLAN with WPA-Enterprise such as eduroam. Further cross-location evaluations are conceivable. The added value of WIDS for recording attacks on monitored networks clearly lies in the procurement of information. How this information is used in further processes depends on the respective area of application, but requires corresponding clarification.

Bibliography

[Be05] Berghel, H., Uecker, J.: WiFi attack vectors. In: Commun. ACM 48 (8), pp. 21-28. 2005.

[Gr66] Green, D. M., Swets, J. A.: Signal detection theory and psychophysics. John Wiley, New York, 1966.

[Gr19] Graylog, Inc: The Graylog Advantage. 2019. URL graylog.org/resources/the-graylog-advantage (accessed 29.04.2022).

[Ha14] Haralson, C.: How To: Create A Fake Access Point On Kali Linux (Rogue AP MItM Attack). 2014. URL youtube.com/watch?v=HePt2J4uSno (accessed 29.04.2022).

[Ka19] Kasongo, Sydney Mambwe; Sun, Yanxia (2019): A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System. In: IEEE Access 7, pp. 38597-38607. 2019.

[Of22] Offensive Security: Kali. The most advanced Penetration Testing Distribution. 2022. URL www.kali.org (accessed 29.04.2022).

[Os18] Öst, A.: Evaluating LoRa and WiFi Jamming. Student thesis. Mid Sweden University. Faculty of Science, Technology and Media, Department of Information Systems and Technology. 2018.

[Ri21] Rittelmeier, H.: Ortung eines WLAN-Clients im Selbstbau: Möglichkeiten und Grenzen. In: Honekamp, W., Povalej, R., Berner, S., Fähndrich, J., Labudde, D. (Eds.): Polizei-Informatik 2021, pp. 19-27

[V117] v1s1t0r1sh3r3 (2017): airgeddon. A multi-use bash script for Linux systems to audit wireless networks. github.com/v1s1t0r1sh3r3/airgeddon (accessed 29.04.2022).

[Wi22a] wigle.net: Statistics. WiFi Networks Over Time. 2022. URL wigle.net/graph-large.html (accessed 29.04.2022)

[Wi22b] wigle.net: All the networks. Found by Everyone. 2022. URL wigle.net (accessed 29.04.2022)

[Ya13] Yang, C., Gu, G.: Security in Wireless Local Area Networks. In: Chen, L., Jiahuang, J., Zihong, Z. (Eds.): Wireless Network Security: Theories and Applications. Springer Berlin Heidelberg, pp. 39-58. 2013.