This full-text paper was peer-reviewed and accepted to be presented at the IEEE WiSPNET 2016 conference.

A Wireless Intrusion Detection System for 802.11 Networks

Zeeshan Afzal*, Judith Rossebø†, Batool Talha‡, and Mohammad Chowdhury†
*Department of Computer Science, Karlstad University, Sweden, zeeshan.afzal@kau.se
†Integrated Operations, ABB AS, Norway, [judith.rossebo, mohammad.chowdhury]@no.abb.com
‡ABB Corporate Research, ABB AS, Norway, batool.talha@no.abb.com

***Abstract*—Wireless local area networks (WLANs) deployment is increasing rapidly. At the same time, WLANs have become an attractive target for many potential attackers. In spite of that, the de facto standard used to implement most WLANs (IEEE 802.11) has what appear to be residual vulnerabilities related to identity spoofing. In this paper, a pragmatic study of two common attacks on the standard is conducted. These attacks are then implemented on test beds to learn attack behavior. Finally, novel attack signatures and techniques to detect these attacks are devised and implemented in a proof of concept Wireless Intrusion Detection System (WIDS).**

I. Introduction

The popularity of WLANs based on the IEEE 802.11 standard has rapidly increased in the last decade. With the advent of Internet of Things (IoT), devices running on the IEEE 802.11 standard protocol are becoming ubiquitous. Unfortunately, IEEE 802.11-based WLANs are prone to a large variety of attacks that attempt to degrade security. This is because, even after putting a lot of effort to make the WLANs secure e.g., by introducing 802.11w, IEEE 802.11 standard still has security flaws and vulnerabilities [4]. Furthermore, the millions of legacy 802.11 devices using the old standards are still insecure. Among the different weaknesses in the 802.11 standard, identity spoofing vulnerabilities and the resulting attacks from them are the topic of discussion of this paper.

A. Background and Motivation

Confidentiality, integrity, and availability (CIA) are three main pillars of security. Finding the optimal balance between these aspects to ensure a more secure WLAN is a challenging problem to address. Protocols such as Wired Equivalent Privacy (WEP) [1], Wi-Fi Protected Access (WPA) [26], and the WPA2 [26] have been devised to cater for the confidentiality and the integrity aspects of WLAN. However, availability is that security attribute, which has been ignored to a great extent. Furthermore, the ability to provide the required confidentiality and integrity is tied to the fact that the data frames in the 802.11 standard are protected by the security protocols. It is in contrast to the protection of management and control frames of 802.11. Meaning thereby, IEEE 802.11's management and control frames are mostly transmitted over-the-air without encryption. Since, the management and control frames are not protected; the fields of these frames (e.g. source or destination address) can be spoofed easily using tools such as [8]. This effectively means that an attacker can request services on behalf of other innocent nodes in the network. Moreover, utilizing identity spoofing the attacker can launch denial-of-service (DoS) attacks to disrupt the availability of the network as shown by previous research [5], [14], [17].

Although the DoS attacks are possible on all OSI layers, lower layers (layer 1 & layer 2) are of more significance in wireless context. Typical security enhancing solutions are firewalls and intrusion detection systems (IDSs). Both of these mechanisms focus mainly at the OSI layer 3 or higher and are largely ineffective at securing the lower layers [24]. Firewalls are commonly integrated in the operating systems of the embedded devices running WLAN protocols to perform basic filtering. On the other hand, IDSs are not commonly exploited in wireless networks. Limited utilization of IDSs in wireless networks is because the WLAN nodes have limited resources to run any heavy-weight security mechanism. Their main goal is to provide wireless functionality for long times.

Designing a security mechanism that is suitable for the resource-constrained wireless environment is the topic of this paper. In this context, the work presented here is inclusive of the development and analysis of an open-source WIDS for 802.11 networks operating in the infrastructure mode [13]. The proposed WIDS does not require to be run on every node in the network which makes it suitable for the wireless environment. Currently, the WIDS is capable of detecting two common Layer 2 attacks in 802.11 networks i.e., the *deauthentication attack* [13] as well as the popular *evil twin attack* [13].

B. Related Work

There have been only a few contributions to intrusion detection at lower OSI layers (layer 2 and below) for wireless networks. Snort-Wireless [24], AirIDS [15], and WIDZ [2] are examples of some potential open source WIDSs. These tools, unfortunately, did not achieve the required maturity level and therefore, have been discontinued. A WIDS is proposed in [18] which is based on the principle of distributed network monitoring [16]. The solution is limited by its dependency on a specific operating system and non-extensibility in terms of the attacks it can detect. In addition, the overhead generated by such distributed approaches is quite high for a wireless environment [25].

Nguyen et al. [20] leveraged the factorization problem and developed a protocol to authenticate the management frames and prevent the deauthentication attack. The solution

978-1-4673-9338-6/16/$31.00 ©2016 IEEE

is promising but the modified device drivers only work for a particular chipset and no open-source implementation was provided. Other solutions [22] and [27] are based on scripts and sequence number analysis respectively, to detect 802.11 attacks. The employed detection methods have significant issues and limitations. The solutions to detect deauthentication attack are not robust and rely either only on a threshold value which could lead to a lot of false positives or detection of odd sequence numbers which can be evaded. The detection of the evil twin attack is limited to specific attack tools which could also be defeated. In [9], authors propose an interesting way to detect identity spoofing in relatively large 802.11 networks with multiple APs. They claim that while malicious nodes can lie about their MAC address, the signalprints they produce are correlated to their physical location and can not be changed. Thus, by observing the difference in signalprints, it is possible to identify spoofed deauthentication requests from malicious nodes and thus detect the deauthentication attack. The work did not provide any validation under real attacks but makes a lot of sense conceptually. The work in [23] describes a machine-learning based approach to detect jamming attacks on 802.11 networks. This work is effective but only concerns jamming attacks which are performed at OSI layer 1 by emitting malicious radio signals. The attacks that we are interested to detect operate at OSI layer 2.

In contrast, there exists some relatively concrete work in literature [7], [11], [25], [29] relating to light-weight intrusion detection specific for peer-to-peer or Wireless Mesh Networks (WMNs). The work in [11] implemented a light-weight anomaly based IDS called OpenLIDS suitable for detecting malicious behaviors caused by malwares. The IDS is demonstrated to be accurate in its detection of threats while still consuming low resources. Rodrigo et al. [7] also proposed and practically implemented an open source IDS (DrogoIDS) for WMNs. Their IDS uses active probing to detect malicious nodes in the network. The IDS only needs to be run on one trust worthy mobile node in the network making it suitable for wireless environment. The IDS is shown to be effective against detecting routing attacks like black hole and colluding misrelay attack [12]. Both of these research works are interesting but only suited to peer-to-peer environment where there is no centralized Access Point (AP). Deauthentication attack and evil twin attack are specific to 802.11 networks operating in the infrastructure mode with a centralized AP. In this context, OpenLIDS and DrogoIDS while architecturally well designed, can not detect these attacks.

C. Novelty and contribution

To the best of our knowledge, we do not know of any practically implemented open source WIDS solution specific for detection of OSI layer 2 attacks such as the deauthentication and the evil twin attack. The related approaches in the literature are either obsolete, limited in their detection method, architecture and/or scope to detect deauthentication and evil twin attacks. The approach by Rodrigo et al. [7] is architecturally close to our work but focuses on WMNs. The novelty of our proposed WIDS solution and a synopsis of the key contributions of this paper is as follows: (1) An open source WIDS suitable for resource-constrained wireless environment; (2) Detailed discussion of the development of the WIDS; (3) Diverse detection logic to detect two OSI layer 2 attacks consisting of signature as well as anomaly based techniques. The detection criteria incorporates more variables than any previous work; (4) A practical implementation of the WIDS with detection of the deauthentication attack and the evil twin attack prototyped; (5) Demonstration of the working of the devised prototype under real attacks; (6) Assessment of the enhanced WIDS using standard metrics; and (7) Evaluation of this enriched WIDS using NSA benchmark requirements.

D. Paper Structure

The rest of the paper is structured in the following way. Section III and Section IV describe the working of the attacks, present the test beds and how they are used to implement the attacks. Section V describes the proposed methodology and the prototype WIDS. Section VI and VII are dedicated to the description of detection logic for both attacks respectively. The testing and evaluation results from the prototype WIDS are described in section VIII. Finally, Section IX provides a conclusion.

II. The Deauthentication attack

Deauthentication frames are subtypes of management frames [10]. These frames are used to gracefully terminate an existing authentication in a 802.11 wireless network operating in infrastructure mode. In a deauthentication attack, an attacker sits between a client station and the access point (AP) to which the client is connected. The attacker waits for the client to authenticate itself to the AP and then injects spoofed (source/destination MAC address) deauthentication frames into the network destined to either party. Once stations receive such frames, they follow the standard and stop the ongoing communication by resetting their connection states [13].

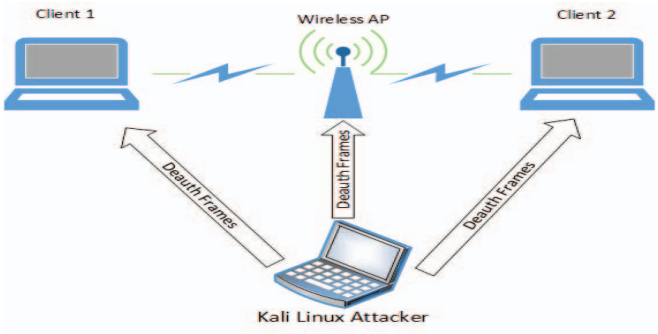

Fig. 1: Test bed for the deauthentication attack.

The test bed (Fig. 1) consists of a wireless network with an AP and two clients. The AP is connected to the internet and provides the service to all the authenticated clients. The attacker machine operates on Kali Linux v1.0.6 [3]. This machine is used to launch the deauthentication attack using

the aircrack-ng tool [8] in the manner defined by its documentation.

After a successful attack, the client stations are disconnected from the network and are not able to connect back until the attack is stopped.

III. The Evil Twin Attack

An evil twin attack [13] consists of an attacker controlled AP which is used to draw clients toward it. The attack is named evil twin because the attacker configures an AP to use the same Service Set Identifier (SSID) in beacon frames as some other active legitimate AP nearby. The reason behind using same SSID is to fool clients. Unsuspecting clients may mistake this evil twin AP as the original AP and connect to it. If clients are tricked to connect to such an evil twin AP, then different malicious activities can be performed because the attacker is now essentially man-in-the-middle.

Figure 2 shows the test bed used to implement the evil twin attack. First, a new AP is started using the airbase-ng module of aircrack-ng tool [8]. This AP assumes the role of a legitimate AP in the testing. Once this AP was up and running, a client (windows machine) was connected to it (displayed as blue lines). Later, a new AP with the same SSID was configured and started to emulate an evil twin of the legitimate AP. The attacker can simply wait for the client to connect to evil twin AP or alternatively can launch a deauthentication attack to break the authentication between the client and the legitimate AP (displayed as red lines). The client will immediately try to reconnect and instead connect to evil twin AP. A key observation during the implementation was that for the attack to be successful, the evil twin AP always needs to have a higher transmit power level compared to the legitimate AP's power level. The notion behind this is that given two resembling APs (same SSID, etc.), a client always prefers an AP with higher transmit power level [21]. After a successful evil twin attack, the evil AP becomes a man-in-the-middle and can eavesdrop on or modify the traffic.

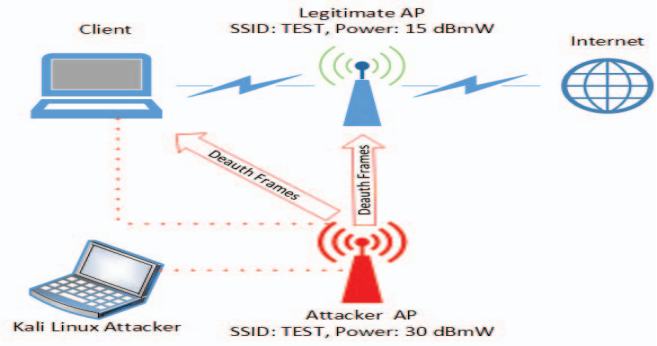

Fig. 2: Test bed for the evil twin attack.

IV. Proposed Solution: The Wireless IDS

This section discusses the proposed WIDS's components and design choices.

A. Components

Figure 3 shows a high level diagram of the WIDS system. The system consists of three interdependent modules namely the collection module, the logging module, and the analysis and detection module. These modules interact with each other to provide detection of wireless attacks.

1) The collection module: The collection module is responsible for collecting the wireless traffic from a wireless channel. It makes use of the wireless network card and operates it in the monitor mode. The module provides support to capture packets according to user supplied filters. In this case, the module is configured to sniff for 802.11 wireless frames only. Once collected, the traffic is passed on to the logging module for handling.

2) The logging module: The logging module takes the traffic from the collection module and logs it for further analysis. It uses the host file system to store the captured traffic. The module takes care that traffic which has been processed is discarded before new traffic arrives to ensure that the log size remains almost constant and does not occupy additional memory and resources.

3) The analysis and detection module: The analysis and detection module is the core module of the whole system. It is responsible for reading the captured traffic, parsing it in understandable form and examining it using the logic discussed in the forthcoming sections to detect attacks. If certain events are detected, it triggers alarms to notify the occurrence of intrusions.

B. Design, Architecture and Deployment

C/C++ is the most commonly used language for applications of embedded devices. However, some embedded systems might have sufficient resources to allow for the usage of a higher level language. In addition, the task of implementing WIDS in C/C++ will be cumbersome for a prototype implementation. Thus, a higher level language i.e. Python is used to implement the proposed WIDS. The proposed WIDS is intended to be deployed on a single node in the network controlled by the administrator. For example, it could run on an AP and protect all the clients in its range. This makes the WIDS ideal for wireless environment where resources of the nodes are limited. The node running the WIDS needs to be equipped with a wireless interface operating in monitor mode.

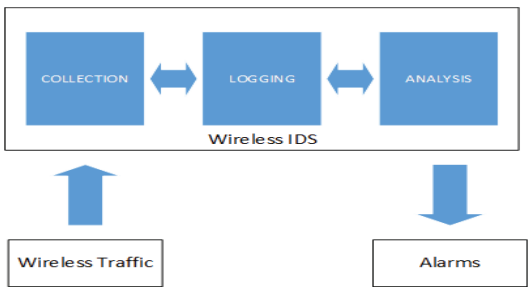

Fig. 3: High level diagram of the proposed WIDS.

Depending on the size of the network, one such node might be enough for an average sized 802.11 network. For large networks, multiple such nodes can be positioned. As in [7], our proposed WIDS also assumes that the node which runs the WIDS is trustworthy and has sufficient computing and energy resources. Once the WIDS is running, it starts capturing 802.11 frames and subjects them to rigorous testing and analysis. The WIDS also performs book keeping. It passively keeps track of all the APs in the neighborhood and logs their SSID, MAC, power and channel number. The detection logic tests traffic with special indicators (discussed in next sections) to decide whether it is benign or intrusive. If attacks are detected then attack specific details are stored in log and alarms are generated.

V. Detection of the Deauthentication Attack

This section describes the indicators used by the analysis and detection module to detect the deauthentication attack.

A. Indicators

The deauthentication frames also have a valid use in the network. It is therefore indispensable to have a detection logic with the ability to filter the deauthentication frames that are part of an attack from non-malicious benign deauthentication frames. If the logic cannot differentiate between these two cases then it will consider every deauthentication frame as part of an attack (as is the case with previous approaches [22] and [27]). This would lead to a high number of false positives. To counter this, we propose the use of three novel indicators (see below: 3, 4 and 5) to detect the attack with low false positives. When these novel attack indicators are combined with the indicators already suggested in literature (see below: 1 and 2), the WIDS is able to detect the attack with a low false positive rate. All indicators are described next.

1) *Number of Deauthentication frames (threshold)*: This is a typical indicator that involves identifying and counting the number of deauthentication frames within a time frame and comparing the count with a threshold value. The threshold value is the maximum acceptable deauthentication frames and is challenging to decide. This threshold will vary from network to network. The optimal value has to be discovered by analyzing the wireless network. The proposed WIDS uses the algorithm proposed in [6] to decide the threshold.

2) *Time span or duration*: Another criterion in the detection is the time window to perform the analysis on. A smaller window, say 10 seconds, will suffice for the prototype. A larger window will prevent timely detection of the attacks.

3) *Number of duplicates (same src MAC to same dst MAC)*: Number of duplicate deauthentication frames can further help in filtering the attack deauthentication frames from the benign ones. Under normal conditions, duplicate deauthentication frames are uncommon in a network during a sampling time. Therefore, the hypothesis that many duplicate deauthentication frames in one snapshot (10 seconds) indicate a flood deauthentication attack is logical.

4) *Reason for deauthentication*: The IEEE 802.11 standard [10] defines different reason codes to identify the reason for which a station was disconnected [28]. During the attack implementation, it was observed that most attack tools use a constant reason code e.g., *code 7* in all the deauthentication frames. This observation of a consistent reason code can be used as an attack indicator. One might argue that the same reason code can also be used by legitimate deauthentication frames, but all legitimate deauthentication frames in one snapshot using the same reason code is not normal. Therefore, all deauthentication frames in a snapshot using a constant same reason code is a good indicator of a deauthentication attack.

5) *Data frames after deauthentication*: Finally, an anomaly based technique is proposed to verify that the detected deauthentication frame is part of an attack and not a benign deauthentication frame. This test is based on the 802.11 standard specification [10] that deauthentication frames are notifications which stop all communication. No data frames should be sent by the communication parties after these frames. However in an attack scenario, since the deauthentication frames are spoofed and injected by an attacker, the detected deauthentication frames do not necessarily mean that stations have already terminated the communication. In fact, the stations could be unaware of these frames and might be sending the regular data frames. Using this knowledge, the detection of data frame/s between two stations after the detection of a deauthentication frame/s sent by one of them signals that the deauthentication frame was spoofed and came from the attacker.

B. Operation

Algorithm 1 shows the steps involved to detect a deauthentication attack. The detection of a deauthentication flood attack can be performed by using indicators 1, 3 and 4 or 5 (lines 6-11). In comparison, a deauthentication attack with fewer frames is hard to detect. In such a case, it is challenging to establish whether the detected frames are benign or injected by an attacker. The anomaly based indicator is used in such a situation (lines 12-14).

C. Robustness of detection

The detection of the deauthentication attack is robust against the tool or method used by the attacker. If its a flood attack (most common scenario), then it can not escape all proposed indicators. An attacker can try to spoof the reason code of every deauthentication frame to evade the reason code indicator. He/She could also try to avoid a flood and send fewer frames. Both attempts will still be detected because of the data frames anomaly indicator since the attacker has no control over it.

VI. Detection of the Evil Twin Attack

This section describes the indicators used by WIDS in the analysis and detection module to detect the evil twin.

Algorithm 1 Detection of Deauthentication attack

Input: Snapshot of Frames
Output: Alarm

```
1:  while frames in snapshot do
2:    read type;
3:    read sub_type;
4:    if type = mgt and subtype = deauth then
5:      get count;
6:      if count>threshold and duplicates then
7:        if reason code = same then
8:          generate alarm;
9:        else if data frames after deauth; then
10:         generate alarm;
11:       end if
12:     else if data frames after deauth; then
13:       generate alarm;
14:     end if
15:   else
16:     return;
17:   end if
18: end while
```

A. Indicators

The evil twin attack relies on scamming clients into connecting to the evil AP. As discussed before, the attack makes use of the AP selection policy in most clients which is based on just the transmit power level. Three novel indicators proposed by us (see below: 1, 2 and 3) to detect such an attack are described next. When these novel attack indicators are combined with an old indicator (see below: 4) previously suggested in literature [22], the WIDS is able to detect the attack effectively. Although this work was carried out prior to a well received recent survey [13], the findings in the study have validated the attack indicators proposed in this work.

1) *Number of beacon frames*: The evil twin attack involves an AP which broadcasts beacon frames to advertise its existence to clients. These beacon frames are counted and compared against a threshold. The threshold is set to almost twice the average number of beacon frames in normal conditions. This threshold level is also validated by the survey [13].

2) *SSID*: Part of the beacon frame is also a name, which is the SSID of the AP broadcasting the beacon frames. This SSID is identical to that of a legitimate AP in the case of an evil twin attack. SSID of every beacon frame is recorded.

3) *Power*: Beacon frames also contain a Received Signal Strength (RSS) value. This RSS value co-relates directly with the transmit power of an AP and can be used to keep track of the transmit power level of an AP. This RSS value is recorded. The survey [13] also established that signal strength is a good criterion to identify such attacks.

4) *Timestamp*: This is a typical indicator which relies on the observation that beacon frames include a timestamp that should increase incrementally with every transmitted beacon frame. However, in an attack scenario, the timestamps are often set to use a constant value.

B. Operation

Algorithm 2 shows the procedure involved to detect an Evil Twin attack using the above mentioned indicators. The WIDS passively keeps track of the APs in the range and knows their average transmit power level. For detection , first of all, the WIDS detects whether a flood of beacon frames exists. Such a flood is detected by checking the count of beacon frames using indicator 1 from above. Based on the observations in [13], we set the threshold to almost double (2x) the average number of beacon frames in normal conditions. This means that if the count of beacon frames exceeds this threshold (line 6), a beacon flood is detected. If a flood is detected, the WIDS checks the timestamp fields of beacon frames in the flood (line 7) using indicator 4. If the time stamps are also constant, then the WIDS generates an alarm. However, if there is no beacon flood (lines 18-23) or there is a beacon flood but timestamps are not constant (lines 9-14), the WIDS relies on the signal strength difference between a legitimate AP and an evil twin AP to detect the attack. It checks whether any AP increased its transmit power significantly (threshold is set to 3 dbmW) from the previously known value (using indicator 3). If such an instant is detected then the WIDS generates an alarm. This is because during a normal and fixed environment, AP transmit power should not change abruptly. Such a symptom indicates that the detected beacon frames are coming from an attacker AP since it has a higher transmit power level than expected which is the prerequisite for an evil twin attack.

C. Robustness of detection

The detection of the evil twin attack is robust against the tool or method used by the attack. An advanced attacker might try to evade the indicators based on detecting a beacon flood or constant timestamps by developing a customized attack tool. In such a situation, the indicator based on power level difference will provide the necessary robustness and be able to detect the attack. This is due to the fact that, to launch an evil twin attack, it is must for an attacker to use a higher transmit power level compared to the legitimate AP. Thus, no matter how the attack is launched, it will get detected.

VII. Evaluation of the Proposed WIDS

This section is dedicated to the evaluation of the proposed WIDS. The section is divided into two main parts. The first part evaluates the performance of the WIDS with respect to some traditionally used security metrics. In the second part, a different approach is adopted to evaluate the WIDS.

A. Evaluation using standard security metrics

For evaluation, a detailed analysis is performed. Different possible outcomes of the WIDS detection, i.e., false positives, false negatives, true positives and true negatives are demonstrated using a confusion matrix analysis. Based on the analysis, a percentage accuracy level of the WIDS is calculated. A confusion matrix is used to depict the information about actual and predicted classifications. These classifications are conducted by the WIDS, which groups every 802.11 frame into one of the two classes, an attack frame or a benign frame.

Algorithm 2 Detection of Evil twin attack

Input: Snapshot of Frames
Output: Alarm

```
while frames in snapshot do
    read type;
    read sub_type;
    if type = mgt and subtype = beacon then
        get count;
        if count>threshold1 then
            if timestamp = constant then
                generate alarm;
            else if SSID in DB; then
                if power > threshold2 then
                    generate alarm;
                else
                    return;
                end if
            else
                return;
            end if
        else if SSID in DB; then
            if power > threshold2 then
                generate alarm;
            else
                return;
            end if
        else
            return;
        end if
    else
        return;
    end if
end while
```

1) Analysis of WIDS under a deauthentication attack: The evaluation is performed to test the detection accuracy of the proposed WIDS under a real deauthentication attack. It is assumed that during the evaluation, no other attack frames are in the wireless medium except the ones injected for testing.

TABLE I: Data for evaluation of detection logic.

	Deauthentication attack	Evil twin attack
Attack instances	250	20
Benign instances	750	10
Total instances	1000	30
Alarms generated	360	22

The data set used in this evaluation is summarized in Table I. To verify that the detected or predicted deauthentication attack frames are actually the ones that were sent from the attack tool, their source MAC address was manually verified using the statistics provided by the WIDS. Based on the results, a confusion matrix analysis is performed to calculate the false negative and the false positive rates. Accuracy level of the detection is also calculated.

TABLE II: Confusion matrix for deauth attack detection.

	Deauthentication attack frames	Benign frames
Deauthentication attack frames	250	0
Benign frames	110	640

Table II shows the results of classification done by the WIDS. All 250 deauthentication frames that were sent from the attack tool were correctly classified by the system as attack frames (true positives). No deauthentication frame were incorrectly classified as benign (false negatives). Similarly, a total 110 benign frames were wrongly classified as attack frames by the WIDS (false positives). 640 benign frames were correctly classified by WIDS as benign (true negatives).

Table IV shows results from the calculation of false negative and false positive rates along with their formulae. False negative rate was found to be 0% which is ideal while false positive rate was 14.6%. Accuracy of WIDS is also calculated by adding up total number of correct classifications done by the WIDS and dividing the sum by the total number of frames that were sent. The accuracy was found to be 89%.

2) Analysis of WIDS under an evil twin attack: Analogous to the evaluation above, the same methodology is used to evaluate the WIDS under an evil twin attack. Table I shows the data used for evaluation.

TABLE III: Confusion matrix for evil twin detection.

	Evil twin attack instances	Benign instances
Evil twin attack instances	20	0
Benign instances	2	8

Table III shows the classification performed by WIDS. A total of 20 evil twin attack instances were detected by the WIDS (true positives) which is exactly equal to the number of attack instances sent to the WIDS. The WIDS detected all instances so there were no false negatives. WIDS wrongly detected 2 other instances of evil twin attack (false positives) and 8 instances were reported as benign (true negatives).

Similar to the calculations done for the deauthentication attack, Table IV also shows results for the evil twin attack. The false negative rate was again found to be 0%, however there was a higher number of false positives (20%). Overall WIDS accuracy was calculated to be 93.3%.

TABLE IV: Evaluation results for both attacks.

Metric	Formula	Deauthentication Attack	Evil Twin Attack
FN rate	FN / (FN + TP)	0%	0%
FP rate	FP / (FP + TN)	14.6%	20%
Accuracy	Correctly classified / Total frames	89%	93.3%

B. Evaluation using benchmark requirements

This part of evaluation is inspired by the NSA document [19] that provides a list of benchmark requirements

that an IDS should ideally meet. Not all the requirements in the document are relevant for a Wireless IDS (operating at OSI layer 2), therefore we shortlisted a total of 13 related requirements. The requirements and whether the proposed WIDS meets them are shown in Table V. All the requirements carry equal weight in the evaluation.

TABLE V: Evaluation using benchmark requirements.

Requirement	Conformed	Score
Monitor all channels in a frequency band	No	0
Provide real time detection	No	0
Detect the actual channel of frames	Yes	1
Geo-locate the source of frames	No	0
Detect signal strength of every frame	Yes	1
Detect DoS attacks	Yes	1
Detect man-in-the-middle attacks	Yes	1
Capture all frames (even corrupted frames)	Yes	1
Keep log size to minimum	Yes	1
Detect protocol anomalies	Yes	1
Diverse detection logic	Yes	1
Customizable attack signatures	Yes	1
Collect statistics	Yes	1
Obtained Score / Maximum Score		10 / 13

C. Summary

In this section, the proposed WIDS was evaluated to verify the effectiveness of the devised detection methods and also identify the weaknesses. For a proof of concept, the detection accuracy of 89% and 93% for the two attacks is good enough. It shows that the proposed attack signatures worked but there is room for more improvement. Specially, the false positive rates can be reduced in future work by polishing the detection logic e.g., by using more indicators. The proposed WIDS also met ten expected features required from an IDS.

VIII. Conclusion

In this paper, we described the concept and practical implementation of an open source WIDS that is suitable for the resource-constrained wireless environment. The WIDS employs novel attack indicators and a diverse detection logic to detect two common OSI layer 2 attacks. Evaluations are performed to evaluate the performance and functionality of the proposed WIDS. The results demonstrate that the detection accuracy is fairly high and the WIDS provides most of the expected features. The modular design of the WIDS allows it to be further improved by reducing the false positive rates and by adding new attack signatures to detect other wireless attacks. The proposed WIDS will be released under an open-access license for the benefit of the whole community.

References

[1] IEEE Computer Society. Wireless LAN MAC and physical layer specifications. *IEEE Standard 802.11, 1999 Edition*, 1999.

[2] Widz, 2002 (accessed Nov 25, 2015). http://freecode.com/projects/widz.

[3] Mati Aharoni, Devon Kearns, and Raphaël Hertzog. *Kali Linux*, 2014 (accessed Nov 25, 2015). http://www.kali.org.

[4] Md. Sohail Ahmad and Shashank Tadakamadla. Security evaluation of IEEE 802.11w specification. In *Proceedings of the 4th ACM conference on Wireless network security*, pages 53–58, 2011.

[5] John Bellardo and Stefan Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the 12th USENIX Security Symposium*, 2003.

[6] Rupinder Cheema, Divya Bansal, and Sanjeev Sofat. Deauthentication/disassociation attack: Implementation and security in wireless mesh networks. *International Journal of Computer Applications*, 23(7), 2011.

[7] Rodrigo do Carmo and Matthias Hollick. DogoIDS: A mobile and active intrusion detection system for IEEE 802.11s wireless mesh networks. In *Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy*, pages 13–18, 2013.

[8] Thomas d'Otreppe. Aircrack-ng, 2015 (accessed Nov 25, 2015). http://www.aircrack-ng.org/.

[9] Daniel B. Faria and David R. Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *Proceedings of the 5th ACM Workshop on Wireless Security*, pages 43–52, 2006.

[10] IEEE 802.11 Working Group. IEEE Standard part 11: Wireless LAN MAC and physical layer specifications. *IEEE Std*, 802, 2010.

[11] Fabian Hugelshofer, Paul Smith, David Hutchison, and Nicholas J. P. Race. Openlids: a lightweight intrusion detection system for wireless mesh networks. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, pages 309–320, 2009.

[12] Bounpadith Kannhavong, Hidehisa Nakayama, Yoshiaki Nemoto, Nei Kato, and Abbas Jamalipour. A survey of routing attacks in mobile ad hoc networks. *IEEE Wireless Communications*, 14(5):85–91, 2007.

[13] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Communications Surveys Tutorials*, PP(99), 2015.

[14] Chibiao Liu and James T. Yu. Rogue access point based dos attacks against 802.11 wlans. In *Proceedings of the 4th Advanced International Conference on Telecommunications*, pages 271–276, 2008.

[15] Michael Lynn. Airids, 2001 (accessed Nov 25, 2015). http://sourceforge.net/projects/airids/.

[16] Amitabh Mishra, Ketan Nadkarni, and Animesh Patcha. Intrusion detection in wireless ad hoc networks. *IEEE Wireless Communications*, 11(1):48–60, 2004.

[17] Lawan A. Mohammed and Biju Issac. Detailed dos attacks in wireless networks and countermeasures. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(3):157–166, 2007.

[18] Jason Murray. An inexpensive wireless ids using kismet and openwrt. *SANS Institute*, pages 8–9, 2009.

[19] National Security Agency. *Guidelines for the Development and Evaluation of IEEE 802.11 IDS*, 2005 (accessed Nov 25, 2015). https://www.nsa.gov/ia/_files/wireless/I332-005R-2005.pdf.

[20] Thuc D. Nguyen, Duc H. M. Nguyen, Bao N. Tran, Hai Trong Vu, and Neeraj Mittal. A lightweight solution for defending against deauthentication/disassociation attacks on 802.11 networks. In *Proceedings of the 17th International Conference on Computer Communications and Networks*, pages 185–190, 2008.

[21] Anthony J. Nicholson, Yatin Chawathe, Mike Y. Chen, Brian D. Noble, and David Wetherall. Improved access point selection.

[22] TJ OConnor. Detecting and responding to data link layer attacks. *SANS Institute InfoSec Reading Room*, 13, 2010.

[23] Oscar Puñal, Ismet Aktas, Cai-Julian Schnelke, Gloria Abidin, Klaus Wehrle, and James Gross. Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation. In *Proceedings of the 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks*, pages 1–10, 2014.

[24] Craig Valli. Wireless snort - A wids in progress. In *Proceedings of the 2nd Australian Computer Network & Information Forensics Conference*, pages 112–116, 2004.

[25] Giovanni Vigna, Sumit Gwalani, Kavitha Srinivasan, Elizabeth M. Belding-Royer, and Richard A. Kemmerer. An intrusion detection tool for aodv-based ad hoc wireless networks. In *Proceedings of the 20th Computer Security Applications Conference*, pages 16–27, 2004.

[26] Wi-Fi Alliance. WPA/WPA2. http://www.wi-fi.org/.

[27] Joshua Wright. Detecting wireless LAN MAC address spoofing. *White Paper, January*, 2003.

[28] Joshua Wright. Weaknesses in wireless LAN session containment. 2005.

[29] Yongguang Zhang and Wenke Lee. Intrusion detection in wireless ad-hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 275–283. ACM, 2000.