An anomaly-based intrusion detection system for IEEE 802.11 networks

Luis Miguel Torres, Eduardo Magaña, Mikel Izal and Daniel Morató
Departamento de Automática y Computación,
Universidad Pública de Navarra,
Pamplona, Navarra, Spain.
Email: luismiguel.torres@unavarra.es

Guzmán Santafé
S21Sec Information Security Labs
Orcoyen, Navarra, Spain.

***Abstract*—The characteristics of Wi-Fi networks and their ever-growing popularity make them an obvious target for attacks. While intrusion detection systems have been popular in wired networks for a long time, their wireless equivalents are very limited. Anomaly-based detection methods have received an increasing interest by the scientific community in the last years. They are able to fight attacks without the need of previous and thorough characterization. However, their application to wireless environments is more recent. In this paper we review some of the last proposals in this field. We also introduce a functional intrusion detection system that combines them in order to offer resilient detection of the most common attacks in 802.11 networks.**

***Keywords*—wireless security; intrusion detection systems; anomaly detection.**

I. Introduction

Nowadays, wireless technologies based on the IEEE 802.11 standards are one of the preferred solutions for local area networks. Unfortunately, this popularity, the fact that they use an easily accessible transmission method, and some vulnerabilities in their link level protocol [1] make them attractive targets for a wide array of security threats. Malicious users may be able to disturb the normal operation of a network with denial of service floods; use MAC spoofing to impersonate legitimate users and gain access to their privileges; monitor and modify even supposedly secure traffic by means of man-in-the-middle attacks; or, simply, listen passively for private information transmitted through the network.

This problem has been traditionally addressed by the inclusion of newer and better security protocols in the different amendments to the standard. WEP, the original 802.11 privacy system, had a vulnerability in its coding algorithm known since 2001 and exploitable since shortly thereafter [2], [3]. Today, a variety of free programs allow obtaining WEP keys easily with different methods. WEP's design flaw was shared by one of its successors, WPA-TKIP, where it has been exploited recently [4]. Fortunately, WPA2 (introduced in 802.11i-2004) is still considered secure although some networks may not implement it because of compatibility issues with older equipment or because it is not practical for some applications (as public hotspots or captive-portal solutions). Moreover, privacy systems, while improving dramatically the security of a network, do not make it immune to all kinds of attacks.

As a consequence, in order to offer an adequate protection to a WLAN, it is necessary to deploy an intrusion detection system (IDS) in the same manner as the ones that operate in wired networks. The offer of wireless intrusion detection systems (WIDS) is, however, scarce and they usually rely heavily on beforehand characterization of the threats, which makes them able to fight well-known attacks but not those that may appear in the future. On the other hand, the newest proposals by the scientific community tend to favour heuristic methods that, albeit introducing more complexity, are able to fight threats known and unknown.

The key feature to monitor in a WIDS is link layer traffic. All frames transmitted in a wireless network share a MAC header, and the MAC protocol defines two types of frames (management and control) that are essential to wireless communications. MAC headers and link layer frames cannot be encrypted (only data payloads may) so they offer the clearest opportunity for security breach and their supervision is thus crucial. Using the information that can be collected from the link layer, different detection techniques have been proposed; some of them will be reviewed in section IV.

In this paper we introduce a WIDS that combines some of these techniques (specifically state and sequence number control) and some new ones in order to offer resilient detection capabilities for a wide range of attacks.

The rest of this paper is organized as follows. Section II introduces basic concepts about intrusion detection. Section III discuses the offer of WIDSs available today. Section IV presents the state of the art in anomaly detection techniques for wireless intrusion. Section V describes the structure and operation of our system. Section VI provides details of the different detection techniques used in it. Section VII presents and discuses the experimental results obtained and section VIII concludes.

II. Intrusion Detection

Attacks to computer networks have been more and more common during the past years making security a major con-

This work has been carried out in collaboration with S21Sec Labs as a part of project Segur@, a CDTI, Ministerio de Industria, Turismo y Comercio de España, financed project inside CENIT program, with reference CENIT-2007 2004.

978-1-4244-9228-2/10/$26.00 ©2010 IEEE

cern. Today it is widely accepted that even in up-to-date and well-configured networks, attacks will occur. It is then necessary to design methods capable of detecting these attacks and fight them before they can do much damage. A system that implements these methods in order to protect a network is known as an IDS. These systems have become increasingly popular especially in wired networks of a certain size.

In general, IDSs work by analysing, in real time, the traffic of the network in search of signs of an attack. They rely on one or various probes situated in key points of the network to sniff all the packets that travel through them. Afterwards, the processing of the captured data can be done in a centralized or distributed way depending on the design of the system.

In order to assess the performance of an IDS two key concepts are used: false negatives and false positives. The former designate occurrences in which the system analyzes traffic corresponding an attack but fails to detect it. On the other hand, in the latter, normal traffic triggers false alarms. The occurrence rates of these two unwanted events are somewhat tied. While increasing the strictness of the detection methods may reduce the rate of false negatives, it probably will increase the rate of false positives.

Detection methods have been traditionally classified by whether they are based on signatures or anomalies. *Signature-based methods* watch for traits of specific attacks (as weird or malformed packets that exploit a known vulnerability, characteristic sequences of packets, or particular data in their payloads) and are able to detect them with very good accuracy. That is to say, they have low rates of false positives and negatives. As a drawback, these methods can only detect attacks that are not only previously known but that have been thoroughly studied and characterized so the signatures need to be constantly updated in order to be effective.

Anomaly detection, for its part, aims to describe what is considered normal in the traffic of a network so that attacks appear as occurrences that deviate of the expected behaviour. As they do not require any knowledge about the threats that they have to fight, they might be able to detect new attacks without any modification. However, they are more complex in their design and tuning and they suffer from higher rates of false positives and negatives. Some authors [5] distinguish a subtype of anomaly detection called *specification detection* although the difference is fuzzy. These methods follow predefined guidelines to detect specific (relevant) anomalies instead of defining what is normal with a previous and more or less automated training. They usually achieve better accuracy by slightly reducing their scope of detection.

Until now, most IDSs have been designed to operate in wired local area networks. As, in wired environments, physical access can be more or less easily controlled and link layer traffic cannot travel between networks, they usually ignore anything below IP level. This allows them to work with different link layer technologies with little modification and makes their operation almost independent of the physical structure of the network they are protecting (as long as they sniff all traffic coming to or leaving the LAN). By a wide margin, the most popular of these systems is Snort [6] up to the point of almost reaching standard status. It is an open-source, signature-based IDS with a very scalable set of rules and an active community that provides timely updates.

III. Wireless Intrusion Detection Systems

The increasing concern in wireless network security, particularly in enterprise environments, has favoured the appearance of a few wireless intrusion detection systems, specially designed to operate in this situation. While traditional IDS can and should be used without problems in these networks to fight attacks concerning the upper layers of the TCP/IP model, wireless networks introduce additional reasons of concern that must be addressed in order to achieve an acceptable level of security. The key difference is that it is extremely easy to access a Wi-Fi network so attacks can and will come from inside. As a first consequence, it is not enough anymore to only monitor the links to the exterior but the WIDS should be able to analyze all the frames between all the stations of the network. Moreover, link layer vulnerabilities can now be exploited by attackers and, given the added complexity and power of the wireless MAC protocols, they will, in fact, be the greatest danger in this case.

Nevertheless, wireless security has not received much attention until recently so the WIDS offer is very limited. There exist some proprietary systems [7], [8] usually distributed and oriented to deployment in large networks. Their intrusion detection strategies are, even though, difficult to assess, as the information published about them is limited. The open-source alternatives are centralized and simpler but also scarce. An extension of Snort, Snort-Wireless, designed for operation in wireless networks and which retained the basic structure and rule engine of the former was, unfortunately, abandoned in 2005. Another open-source IDS, Bro [9], has also been adapted for 802.11 operation [10] using both anomaly and signature-based detection methods, but the modified program does not seem to have been published. Therefore, the only usable alternative is Kismet [11], even if its WIDS capabilities are limited at best. Kismet was originally designed as a wireless network detector and sniffer but it has since incorporated some intrusion detection features. It implements a very small set of simple fingerprint and trend rules but does not offer a way to expand it (as both Snort and Bro do).

IV. Detection techniques

In this section we focus on some of the anomaly and specification based intrusion detection methods proposed by precedent works. In Wi-Fi networks, anomaly-based methods usually supervise general traffic variables (*i.e.* frame size, interarrival time, etc.) whose expected values have been established by a previous empirical training. Specification-based methods detect occurrences that deviate from the 802.11 MAC protocol expected behaviours so the rules they use are inferred from what is to be expected according to the standard.

The system we present in this paper takes into consideration some of the guidelines for designing a WIDS proposed by

Lapiotis *et al.* [12]. It is also inspired by Fayssal, Hariri and Al-Nashif [13] in that we use a wide array of different features for attack detection amongst which the principal two are the following.

A. Transition models

In 802.11 networks, stations must complete an authentication and association process that binds them to an access point before they can start receiving and sending data. This process is simple in unprotected or weakly protected (WEP) networks and more complex when 802.1X authentication (WPA2) is used. In any case, the station must exchange a number of management frames with the AP in a specific order and go through a series of states from unauthenticated/unassociated to fully participating in the network. Given that some attacks may result in illegal transitions between these states, Gill *et al.* [14] have proposed monitoring this feature in order to detect threats such as impersonation or DoS attacks. They implement the full state machine of 802.1X authentication and use an algorithm that detects illegal state transitions, which is very similar to the one we use and will be explained in section 6.

B. Sequence number

Both 802.11 management and data frames carry a sequence control field in their MAC headers that is used for controlling the fragmentation and reassembly processes at link level. A subfield of it, the sequence number, is taken from an internal 12-bits counter (0-4096), which all Wi-Fi stations must have, and which is incremented for each frame transmitted. As a consequence, theoretically, each frame that a station transmits should carry a sequence number one unit greater than the one in the immediately preceding frame.

The sequence number can be used to detect impersonation attacks because even if spoofing a MAC address is relatively easy, the incoherences that will arise between the sequence numbers of the original and spoofed frames are almost completely unavoidable by the attacker. Unfortunately there is a drawback to this application as the sequence number progression is frequently disturbed by frame loss and retransmission (which are both common in wireless environments). Wright [15] proposed a detection method that monitored the sequence number of the frames coming from every station in a network. He addressed this drawback by setting fixed thresholds that distinguish when the difference between the sequence numbers of consecutive frames was abnormal. Dasgupta *et al.* [16] improved this idea by using a fuzzy logic system able to calculate the thresholds by means of a previous training with normal traffic of the specific network. Both proposals are, perhaps, too simple as frame loss is difficult to model and false positives and negatives can be frequent.

Guo and Chiueh [17] introduced a more complex method based on the observation that the probability distribution of the difference between the sequence numbers of two consecutive frames is highly concentrated around one. That is to say that even though retransmissions and frame loss cause that this difference is not one in a sizeable number of cases, the deviation is almost never large. With this in mind they established an algorithm with very restrictive initial thresholds (any difference out of the [-3,2] range was considered anomalous) that, when exceeded, triggered a verification state. In this second phase, the objective was to distinguish if the big gap between the sequence numbers was consequence of massive frame loss (in this case the sequence progression would continue normally after the gap) or if there were two different stations with their different and out of sync counters using the same physical address.

V. S^{2}WIDS: STRUCTURE AND OPERATION

Our system is called S^{2}WIDS (for *Sequence and State based Wireless Intrusion Detection System*). Programmed entirely in C in order to achieve the necessary efficiency for operation in real time, it is designed to read frames from a PCAP interface [18]. It is a centralized system intended to receive data captured by a sniffer near or embedded into the access point of a Wi-Fi network, but it could be adapted to read frames from different sources in bigger networks.

For every frame captured, three different processes are invoked sequentially. A diagram of this is shown in figure 1.

- Frame data parsing: General data about each frame (size, capture time, etc.) and its link header are parsed into a data structure that will be used by the rest of the program. Only 802.11 management and data frames are processed because control frames are almost irrelevant for intrusion detection and discarding them from the beginning reduces overhead.
- Station state update: For each station detected, S^{2}WIDS maintains a series of state variables, counters and timers that must be updated when a frame concerning said station is captured. The number and nature of this variables is highly influenced by the multidisciplinarity of the detection engine as will be explained in section VI.
- Event handling: If a strange event is detected during the state update process, the corresponding handling function is called. This function will study the anomaly in the light of the affected station historic data and decide if it is necessary to raise an alarm. Both those events that are considered dangerous (they produce alarms) and those that are simply odd are stored in different logs for later study.

VI. ANOMALY ENGINE

S^{2}WIDS implements a combination of various detection techniques that describe a series of events and circumstances considered anomalous so that alerts are raised when they are encountered. These events can be roughly classified in five groups.

A. General supervision

This group deals with the interactions between stations (STA) and access points (AP) in general. The events that rise alarms are the following:

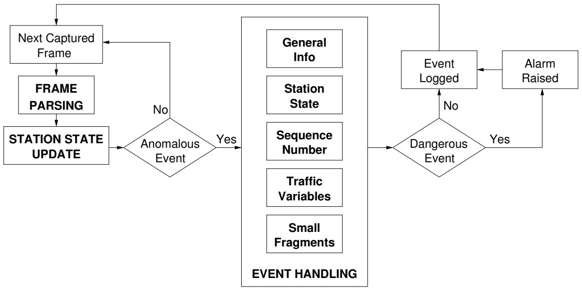

Fig. 1. Structure of S^{2}WIDS

- A STA changes its AP illegally.
- An AP sends frames to a STA that is binded to another AP.
- A STA probes networks but never participates.

Generally, stations that probe but do not participate are not a threat. However, when dealing with passive attacks, this may be the only sign of their presence and provide us with the necessary information to apply detection techniques like the one presented by Yu-Xi *et al.*[19].

Additionally, S^{2}WIDS allows for the use of blacklists and whitelists for both stations and access points.

B. Station state supervision

As was previously explained, stations in a 802.11 network must authenticate and associate with an access point before they can transmit and receive data. These processes are governed by management frames that can be monitored for strange behaviours. We consider the simplest state machine (*i.e.* without 802.1X authentication) which has the following states:

1) Unauthenticated/Unassociated
2) Authenticating (authentication sent from STA).
3) Authenticated (authentication sent from AP).
4) Associating (association request sent from STA).
5) Associated (association response sent from AP).
6) Transmitting data (data frame sent by STA or AP).

When an unexpected change of state happens, it triggers one of the following events:

- Negative shift: a STA falls back to a lower state (produced by deauthentication and dissasociation frames). If frequent, it can be a sign of a DoS attack.
- Positive shift: a STA jumps from a lower to a higher state without going though the ones in the middle. Normally it is caused because some frames have not been captured by the sniffer but it can also be a sign of spoofing.
- Same-state shift: a STA receives or transmits a management frame that should lead it to the state that it currently occupies. Although it can be a sign of both spoofing and DoS, it is usually related to false authentication attempts (or to normal frame loss and retransmissions).
- Unexpected frame: a management frame concerning one STA is captured and this frame is meaningless given the state the STA occupies.

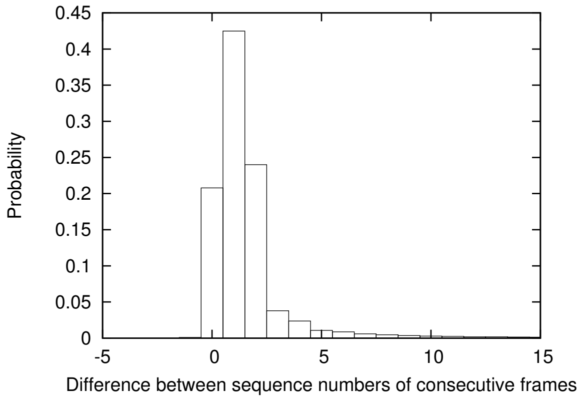

Fig. 2. Probability distribution of the difference between sequence numbers of consecutive frames (aggregated traffic)

- Positive shift after negative shift: a STA experiments a positive shift shortly after a negative one. This is a stronger sign of spoofing than just the positive shift as most spoofed stations are previously disassociated or deauthenticated.

The last two events generate alarms if they are detected as they should be very rare in normal conditions. For the first three, the number of occurrences in a time interval necessary to raise an alarm can be configured. In any case, retransmissions are distinguished by the corresponding flag in the 802.11 header and frame loss is partially detected by monitoring the sequence number so S^{2}WIDS is able to rule out most of the normal occurrences beforehand.

C. Sequence number supervision

As stated in section IV-B, the sequence number in the IEEE 802.11 header is useful in order to detect spoofing attacks. S^{2}WIDS supervises the sequence number in the frames of all the stations in a network searching for incoherences in the expected progression. We have observed a distribution of the difference between the sequence numbers of two consecutive frames very similar to the one presented by Guo and Chiueh [17]. However, it is necessary to previously address the cases of null-function and quality of service (QoS) data frames. The sequence number of the former may be set to any value so they should not be used. The latter take their sequence numbers, according to the standard, from a different counter for each traffic type (TID, Traffic IDentifier). This makes using them for detection purposes more complex and less effective so S^{2}WIDS does not consider them either. Without the effect of those frames, the distribution of the difference between the sequence numbers is shown in figure 2.

As our results are similar, we use, essentially, the same algorithm although we have introduced two lower thresholds for the difference of sequence numbers. One of them, slightly more relaxed than the original, is for those frames of which the retransmission flag is activated (and thus are more likely to be, indeed, retransmissions). The other, more strict, for those that not being retransmissions may have been transmitted or captured out of order. With this scheme, the following anomalous events can be detected:

- Retransmitted frame without retransmission flag: this event is triggered when a frame with its retransmission flag set to zero has a sequence number that suggest that it is a retransmission and falls inside the more restrictive threshold. An alarm is not raised as this can happen normally but it might be worth of particular study.
- Station loses multiple packets: a station has entered the verification state multiple times but the cause seems to be only frame loss. This is not an alarm either but informs that the station might be almost out of the range of the sniffer so a lot of its frames will not be captured and most detection strategies will not work properly.
- Two counters for the same physical address: the verification state has decided that there are, in fact, two different counters for the frames coming from a physical address. This is a sign of spoofing and rises an alarm.

D. Small fragments supervision

Fragmentation attacks are used to break WEP and WPA-TKIP keys by means of frame fragments of increasing size. In these attacks, the first fragments transmitted are remarkably small. However, when a frame is normally fragmented at link layer (something quite extraordinary on its own as Wi-Fi MTU is bigger than the one of Ethernet) the first fragments should be the size of the MTU and only the last should be smaller. Detecting this anomalous fragments is possible and S^{2}WIDS does so to fight this kind of attacks.

E. Traffic variables supervision

While the previous detection techniques aim to fight more or less subtle threats as spoofing, some attacks have a huge impact in the normal traffic of a station or even on the traffic of the whole network. Although searching for anomalies in general traffic variables in 802.11 networks is complex, because network traffic is very unsteady and difficult to model, these attacks are very noisy and easy to detect with this method. S^{2}WIDS keeps, for each station, updated statistics such as:

- Number of transmitted data frames (per second).
- Number of transmitted/received management frames (per second)
- Data traffic throughput.

Using them, the following anomalous circumstances can be detected:

- Heavy flow of management frames coming from a station: that station might be carrying out a DoS attack.
- Heavy flow of management frames directed to a station: that station might be suffering a DoS attack.
- Heavy flow of small data frames from a station: that station might be injecting ARP packets into the network in order to break a WEP key [20].

VII. Results and discussion

A. Tuning

In order to tune the detection engine of S^{2}WIDS we used traffic captured in real networks. We resorted to the CRAWDAD repository [21], from where we chose two sets of traces captured from Wi-Fi networks in the 2005 IETF meeting and the SIGCOMM 2008 conference respectively. We worked under the assumption that those networks had not suffered attacks during the conferences. Therefore, we set the various thresholds that the system uses for detection to values that minimized the number of alarms produced, which, as for our previous assumption, were considered false positives.

We found that frame loss is the most common cause of false positives. This is not surprising as S^{2}WIDS relies on the captured frames to update the state of the stations it monitors. When the sniffer fails to capture an important number of frames, the information the system has is not complete and normal occurrences can mimic attacks. Moreover, when the frame losses do not happen only in the sniffer but through all the network (*e.g.* in noisy environments or when some stations are too far from the access points), some anomalous events are also more frequent. For example, if the system captures a dissassociation frame, it will suppose that the station it was addressed to has been dissassociated, but it could have not received the frame and continue sending data to the network. As previously stated, while looser thresholds can be chosen to deal with these frame loss issues, some attacks may then be able to avoid the detection algorithms.

B. Testing

As far as we know, there is not a public set of wireless traffic traces with known attacks on them that can be used to test a WIDS. This would be interesting in order to compare the detection capabilities of different systems. However, attacks specific to wireless networks are not as common as other attacks, so it is more difficult to capture them while they are taking place on a real network. Therefore, we used a Wi-Fi testbed for assessing S^{2}WIDS detection capabilities. It consisted on four PCs accessing Internet (with different applications: web navigation, file downloads, P2P and video streaming) through an AP, and two others that acted as attacker and detection system respectively. Because all the equipment was in the same room, frame loss was less problematic than in the previous settings. The system was, nevertheless, tuned to minimize false positive rates in real networks (as shown in the preceding section) because our objective was to prove if it was able to detect attacks in those conditions.

As a first step we simulated normal traffic without any attack. As expected, S^{2}WIDS did not generate any alarms. Afterwards, we carried out several different attacks (most of them using Aircrack-ng suite [20]). As it is shown in table I, the system was able to detect them all.

Some attacks were detected by two of the detection methods which makes the system more resilient. Moreover, ARP injection and fragmentation attacks are usually used in conjunction

TABLE I
TESTED ATTACKS AND SYSTEM RESPONSE

Attack	**Detection method**	**Alarm**
False authentication	Station state supervision	Too many same-state shifts
MAC spoofing	Sequence number supervision	Two counters for the same physical address
MAC spoofing (with previous DoS)	Station state supervision Sequence number supervision	Positive shift after negative shift Two counters for the same physical address
ARP injection	Traffic variables supervision	Heavy flow of small data frames from a station
Fragmentation Attack	Small fragments supervision	Small first fragment
DoS by deauthentication or dissassociation	Traffic variables supervision	Heavy flow of management frames from STA (attacker) Heavy flow of management frames towards STA (victim)

with false authentications or MAC Spoofing in order to make the AP accept the attacking frames so they will be also detected by more than one method.

Overall, the system is fast being able to process more than 100.000 frames per second in a Pentium IV (3.4 GHz) PC. Real time operation is, therefore, possible with moderate computational requirements.

VIII. CONCLUSIONS

Wireless networks have inherent security problems that make them more prone to be attacked than their wired counterparts. 802.11 standards have addressed some security issues with privacy systems like WPA2. Nevertheless, those systems do not grant immunity to all present attacks and additional vulnerabilities may be found in the future. There is a need of tools that allow detection of those attacks when they happen so that measures can be taken to neutralize the threats. However, wireless intrusion detection is still a field scarcely studied.

In this paper we address this need by introducing a wireless intrusion detection system called S^{2}WIDS. It is an anomaly-based system that implements a multidisciplinary approach to detection using some of the different proposals that the scientific community has provided in the last few years (specifically, state and sequence number supervision). It has been designed to detect the most common attacks in wireless environments and, as it is anomaly-based, it may be able to fight some of the new threats that might arise in the future.

In our tests, S^{2}WIDS has been able to detect all the attacks we have tried, in most cases with two different methods which adds resilience to the system. A previous tuning with normal traffic of the network that is going to be supervised is necessary, nevertheless, in order to minimize false positives caused by frame loss.

Some future improvements may increase the detection capabilities of the system. Currently, S^{2}WIDS centers its detection strategies on the state of the stations but additional information about the AP might be useful to detect threats such as Rogue APs. Better knowledge of the effects of an attack in the statistical characteristics of traffic of a wireless network could also be used to improve the corresponding part of the detection engine. Finally, implementing a detection technique for passive attacks like the one mentioned in section VI-A would also offer a more complete protection.

REFERENCES

[1] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proceedings of the 12th USENIX Security Symposium, Washington DC, U.S.A.*, 2006.

[2] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," in *Selected Areas in Cryptography*. Springer Berlin / Heidelberg, 2001, pp. 1–24.

[3] A. Stubblefield, J. Ioannidis, and A. D. Rubin, "Using the Fluhrer, Mantin, and Shamir attack to break WEP," 2001.

[4] M. Beck and E. Tews, "Practical attacks against WEP and WPA," Cryptology ePrint Archive, Report 2008/472, 2008.

[5] P. Uppuluri and R. Sekar, "Experiences with specification-based intrusion detection," in *RAID '00: Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*. London, UK: Springer-Verlag, 2001, pp. 172–189.

[6] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX conference on System administration, Seattle, U.S.A.*, 1999.

[7] Motorola, "Airdefense solutions," http://airdefense.net.

[8] Airtight, "Spectraguard," http://www.airtightnetworks.com.

[9] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, no. 31(23-24), pp. 2435–2463, December 1999.

[10] R. Neumerkel and S. Groß "A sophisticated solution for revealing attacks on wireless LAN," in *Proceedings of the 3rd International Conference on Trust, Privacy and Security in Digital Business (TrustBus'06), Krakow, Poland*, 2006.

[11] "Kismet," http://www.kismetwireless.net.

[12] G. Lapiotis, B. Kim, S. Das, and F. Anjum, "A policy-based aproach to wireles LAN security management," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, Athens, Greece*, 2005.

[13] S. Fayssal, S. Hariri, and Y. Al-Nashif, "Anomaly-based behavior analysis of wireless network security," in *Proceedings of the 4th IEEE International Conference on Mobile and Ubiquitous Systems: Networking & Services, Philadelphia, U.S.A.*, 2007.

[14] R. Gill, J. Smith, and A. Clark, "Specification-based intrusion detection in wlans," in *Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual*, December 2006, pp. 141–152.

[15] J. Wright, "Detecting wireless LAN MAC address spoofing," www.uninett.no/
wlan/download/wlan-mac-spoof.pdf.

[16] D. Dasgupta, F. Gonzalez, K. Yallapu, and M. Kaniganti, "Multilevel monitoring and detection systems (MMDS)," in *Proceedings of the 15th Annual Computer Security Incident Handling Conference, Ottawa, Canada*, 2003.

[17] F. Guo and T. cker Chiueh, "Sequence number-based MAC address spoof detection," in *Proceedings of the 8th International Symposium on Recent Advances in Intrusion Detection (RAID'05), Seatle, U.S.A.*, 2005.

[18] "Tcpdump/libpcap," http://www.tcpdump.org.

[19] Y.-X. Lim, T. Schmoyer, J. Levine, and H. Owen, "Wireless intrusion detection and response," in *Proceedings of the 2003 IEEE Workshop on Information Assurance, New York, U.S.A.*, 2003.

[20] Aircrack, "ARP request replay attack," http://www.aircrack-ng.org.

[21] D. Kotz and T. Henderson, "CRAWDAD: A community resource for archiving wireless data at Dartmouth," *IEEE Pervasive Computing*, vol. 4, no. 4, pp. 12–14, 2005.