

# Passive Listening and Intrusion Management in Commodity Wi-Fi Networks

Liran Ma, Amin Y. Teymorian, Xiuzhen Cheng

Department of Computer Science  
George Washington University,  
Washington DC, 20052, USA.

Email: {lrma, amin, cheng}@gwu.edu

**Abstract**—We examine a widely accepted myth about passive listening in wireless networks, and give a detailed description of how to achieve real “passive listening.” Then we develop a lightweight intrusion management system called LIMS for commodity Wi-Fi networks. Our system employs novel techniques to increase network resilience to passive and active attacks that could reveal the WEP/WPA-PSK shared key. LIMS has the following nice properties: i) it requires neither specialized hardware nor modification to existing security protocols (e.g., WEP and WPA); ii) the proposed mechanism can be integrated with an access point in a plugin manner; iii) it provides a cost-effective security enhancement to Wi-Fi networks by incorporating free but mature software tools; iv) it has the ability to prevent a certain class of DoS attacks.

## I. INTRODUCTION

The ubiquity of wireless networks has provided the network security field with challenges never seen before. Commodity Wi-Fi networks are particularly vulnerable due to factors such as open medium, cooperative algorithms, inefficient software implementations, potential for rogue hardware, and improper configurations. The traditional way of protecting networks with firewalls and encryption is no longer sufficient and effective.

In current commodity Wi-Fi networks, the most common security protocol, Wired Equivalent Privacy (WEP), has been shown to be breakable even when correctly configured [1]–[3]. Specifically, WEP fails to achieve any of the fundamental security goals of confidentiality, integrity, and availability.

Wi-Fi Protected Access (WPA), which began shipping in 2003, was created in response to serious weaknesses that researchers found in WEP. It serves as a compromise between the necessity of improved security and the restrictions of the legacy WEP hardware. However, WPA does not necessarily work with the first generation access points (APs). Even though the Temporary Key Integrity Protocol (TKIP) is adopted, it still relies on the RC4 encryption algorithm.

When operating in WPA Pre-Shared Key (PSK) mode, a strong passphrase is required. Otherwise, the secret key might be discovered by launching a brute-force dictionary attack on authentication frames. In this attack, the hash of each word in a dictionary is compared to the hashed passphrase used during

the handshake. Of course, if the passphrase used for the WPA encryption is not located in the dictionary, this attack will fail.

IEEE 802.11i (WPA2) enhances the security of the 802.11 standard through the use of AES-based CCMP (Counter Mode CBC MAC Protocol) and 802.1X. But, it is not compatible with legacy systems, and an equipment upgrade would introduce a significant amount of overhead. Moreover, vulnerabilities in 802.11i have already been discovered [4], [5].

Wireless network cards that are capable of capturing all 802.11 transmissions are readily available in the form of commodity Wi-Fi products. All that an attacker must do is to find the appropriate program such as “iwconfig” in Linux to enable this feature. Specifically, the attacker needs to have a network card configured to “passively” capture all encrypted traffic. Once enough relevant packets are captured, the process of obtaining the key is trivial [6]. On the other hand, a type of active attack “injects” packets into the network. This requires a network card that is capable of constructing raw 802.11 frames. Currently, such a card can be purchased for about US \$30 on eBay.

With the advances in hacking software, a person with a limited security background can easily crack a WEP or a WPA-PSK protected Wi-Fi network. The prevalence of some Linux Live CDs such as Backtrack [7] make a large collection of easy-to-use and powerful wireless cracking tools readily available. One example of a common WEP/WPA-PSK key cracking tool is Aircrack-ng [8].

Note that it is possible to obtain the key by only passively listening to the network traffic. A general assumption is that passive listening can be achieved by using a network card in promiscuous mode coupled with a properly configured firewall. In this way, the presence of a device on the network can be prevented from being revealed. Nonetheless, as shown in this paper, that assumption allows a “passive listener” to be detected.

The contributions of this paper are twofold. First we dispel the above myth by giving a detailed description of how to achieve real “passive listening” from a practical perspective. Secondly, we develop a lightweight intrusion management system (LIMS) that targets commodity Wi-Fi networks. LIMS includes three major components: a packet collector, an intrusion preemption engine, and an intrusion detection engine. The proposed system can be connected to or implemented on APs

as small plugins. It works in conjunction with current security protocols like WEP and WPA, and it does not require any specialized wireless equipment.

The rest of this paper is organized as follows. Section II discusses related work. In Section III, detailed descriptions of techniques used to achieve real “passive listening” are provided. The proposed lightweight intrusion management system is elaborated in Section IV. Finally, our conclusion and future research directions are presented in Section V.

## II. RELATED WORK

A typical intrusion detection system (IDS) scans network traffic and generates an alert when an intrusion has been detected. Responses from an IDS can help minimize the amount of damage caused by an attack. When used with intrusion prevention techniques such as realtime traffic flow analysis and automatic attack prevention, the security of a network is further enhanced.

Many IDS techniques have been developed for wired networks. However, these mechanisms cannot be directly applied to wireless networks [9]. Thus, IDS techniques for wireless networks have recently received much attention from the research community.

Zhang *et al.* [9] carefully study the vulnerabilities of wireless networks and propose a security architecture for anomaly detection in mobile ad-hoc networks. In this work, intrusions are detected based on both local and cooperative detection engines.

In [10], an architecture for monitoring enterprise wireless networks using APs, mobile clients, and dedicated sensors is proposed. One of its main objectives is to detect the presence of rogue wireless devices like unapproved APs.

Additionally, Yeo *et al.* [11] improve the performance of wireless monitoring by merging packet captures from multiple network sniffers and carefully selecting sniffer placement. The techniques are exploited to characterize MAC layer traffic and perform retrospective diagnoses. In contrast, LIMS is not limited to layer 2 traffic. It also supports automatic preemption and detection of various network attacks.

Multiple network sniffers are also used in [12] for detecting rogue APs and eavesdroppers. Each sniffer has three network cards, and the intrusion detection capabilities are stymied by MAC address spoofing. LIMS provides techniques to detect rogue clients with spoofed MAC addresses without relying on heavily equipped sniffers. It can also detect sophisticated eavesdroppers that are capable of real “passive listening.”

Characteristics and network usage statistics of IEEE 802.11 WLAN in various settings are examined in [13] and [14]. Such information is important in assisting with the prevention and detection of denial of service (DoS) attacks.

Bellardo *et al.* [15] propose simple schemes based on cooperation among wireless clients to counter various DoS attacks. Changes to the IEEE 802.11 standard are required in some cases. In LIMS, the detection of DoS attacks is made independently of other nodes, and no changes to existing standards are needed.

Aside from detecting passive and active attacks, techniques for securing networks from greedily behaving nodes have

also been explored. An AP-based system called DOMINO is proposed in [16] for the detection and identification of greedy wireless clients. This scheme places the entire computation and storage overhead on the APs, which may have limited CPU and memory resources. Contrary to DOMINO, our scheme distributes its overhead over three computing modules. In [17], attention is paid to specifically detect the MAC layer misbehavior of selfish hosts. With some modifications to the IEEE 802.11 standard, the proposed scheme can simplify the detection of such hosts.

There also exist several commercial products in the area of corporate Wi-Fi security. AirDefense [18] is an example of such products. It uses a combination of radio frequency sensors and a IDS/IPS server appliance to capture, process, and correlate network events. However, the latest release, AirDefense 7.2, has a starting price of US \$7,995.

Our proposed solution is different from previous work in that it is the first intrusion management system that targets commodity Wi-Fi networks from a practical perspective. LIMS improves the resilience of Wi-Fi networks through an elegant coupling of intrusion preemption and detection. Moreover, the mature techniques and freely available software that LIMS employs make it a very efficient and cost-effective solution. Lastly, modifications to the underlying wireless standard are not necessary with LIMS.

## III. PASSIVE LISTENING

Passive listening generally refers to capturing or “sniffing” all packets on a network, especially those not addressed to the device doing the sniffing. It is easier to sniff traffic on a wireless network than a wired one because of the broadcast nature of the medium. That is, a passive listener does not need to have a physical connection to the network. All that is needed is to be within transmission range of the AP.

A more comprehensive type of passive listening involves tuning the wireless card to different frequency bands being used. The network card is instructed to listen to each channel for a fixed period of time or number of messages. The sniffed frames from each channel can be used to launch various attacks.

An attacker is often interested in passively collecting all traffic on a wireless network without being detected. There is a well-accepted myth concerning passive listening: a network card in promiscuous mode and a properly configured firewall cannot reveal the presence of an attacker. We will see in the following section that such “passive listening” is not actually passive.

### A. What Makes “Passive Listening” Not Passive?

In promiscuous mode, a device might respond to packets that are not addressed to it. An example of a packet that can elicit such a response is an ICMP Echo Request (a ping message). Even though the request is addressed to a different device, the eavesdropping device will still generate a reply. This reply contains the eavesdropper’s MAC address, which does not correspond to the IP address of the intended recipient of the request. If an AP detects this unwarranted response, it

can be sure that the origin device is in promiscuous mode. Experienced sniffers can prevent this by setting their firewalls to block all ICMP traffic.

Interestingly, there are still other ways that a passive listener becomes active. Since firewalls filter at the IP layer and above (e.g., at the transport layer to keep state), not all traffic can be blocked. Examples of protocols that generate such traffic include:

- the Address Resolution Protocol (ARP), a protocol that is primarily used for translating IPv4 addresses to Ethernet MAC addresses.
- an extension to ARP called Inverse Address Resolution Protocol (InARP) that performs the inverse of ARP in Frame Relay networks.
- the Bootstrap Protocol (BOOTP), which allows a client to obtain an IP address automatically during the bootstrap process.<sup>1</sup>

As shown in the following subsection, real “passive listening” can still be achieved.

### B. Methods to Achieve Real “Passive Listening”

In this subsection, we discuss two approaches for achieving real passive listening. The first technique involves disabling some options on a network card, while the second relies on a minor modification to the source code of the TCP/IP stack.

1) *Reconfigure Network Card*: One way to prevent responding to any message that uses IP related information<sup>2</sup> is to turn off the TCP/IP stack on a wireless card. This can be done by bringing up the card with no IP address configuration. Since there is no IP information available, no IP packets will be sent. However, the network card, as a device, can still be controlled by the OS for collecting frames in the air. If IP functionalities are desired, the network device can be restarted after the configuration file is changed back.

2) *Recompile TCP/IP Stack*: In a wired network, an eavesdropper could physically cut the transmit wires in a network cable to ensure that messages are never sent. Although the above technique will not work in a wireless environment, modifying the source code of the TCP/IP stack can produce a similar effect.

The “send()” function in the TCP/IP stack is responsible for sending packets. By simply disabling the “send()” function and recompiling the stack, it will no longer be able to transmit packets. Once the source code has been modified, the new TCP/IP stack can be reloaded into the kernel. We note that recompiling a TCP/IP stack might prove to be a time consuming task. Additionally, it could be an inconvenience in environments where frequent changes in networking functionality are needed.

Other network protocol stacks may also reveal the presence of a network device. One such protocol stack is Internetwork Packet Exchange/Sequence Packet Exchange (IPX/SPX), which is used by the Novell Netware operating system. IPX

and SPX are the counterparts to the IP and TCP layers, and IPX can also be transmitted over Ethernet. Similarly, the Network Basic Input/Output System (NetBIOS) protocol can reveal network presence. This protocol assigns each computer on a LAN a NetBIOS name and an IP address in order to allow applications on separate computers to communicate. Consequently, a network card needs to have the above protocols disabled as well.

### C. Why Not Remain Passively Listening?

Although it is possible to passively listen without being detected, doing so may not always be in an attacker’s best interest. The motivation for this choice is attributed to the length of time an attack takes. In particular, it is conceivable that cracking a WEP key with passive listening can take a few hours to several days depending on how busy a wireless network is.<sup>3</sup>

Similarly, a passive attack on a WPA-PSK protected network requires observation of the authentication frames transmitted between a client and the AP. The time spent waiting for an authentication to occur decreases the efficiency of a passive listening attack. Considering an attacker with the intention of getting free Wi-Fi access, active attacks become appealing. For a detailed survey on both types of attacks, we refer the readers to [2], [3], and [19]. Our response to both passive and active attacks is LIMS.

## IV. A LIGHTWEIGHT INTRUSION MANAGEMENT SYSTEM

The type of Wi-Fi network that we consider uses WEP or WPA in conjunction with MAC address filtering. Additionally, we try to avoid rekeying activities, as they require significant overhead. An example of such a network is the one used by the Department of Computer Science at The George Washington University. Although, there are about 20 to 30 active users daily, there are over 600 registered users. As a result, changing the shared secret key would require a large amount of effort.

A seemingly easy way to thwart intruders on these types of networks is to upgrade the security protocols to WPA2. However, WPA2 is not compatible with the legacy hardware that is widely used in commodity Wi-Fi networks. Since large-scale equipment upgrades would incur high costs, simple intrusion preemption and detection plugins for AP devices become attractive solutions. To this end, we propose a lightweight intrusion management system called LIMS that improves the resilience of WEP/WPA-PSK enabled commodity Wi-Fi networks.

Accordingly, LIMS is designed to monitor network activity, respond to events that could lead to disclosure of the secret key, and block unauthorized Internet usage. The three main components that constitute the LIMS architecture are: a packet collector, an intrusion preemption engine, and an intrusion detection engine. An illustration of the overall architecture of LIMS can be seen in Fig 1.

These components can be implemented on an AP or on separate devices that are connected to the AP in a plugin

<sup>1</sup>Although this protocol operates at the IP/Transport layer, it can reveal an attacker’s real MAC address before a spoofed address can be set.

<sup>2</sup>These include both direct and indirect use of IP information, such as a ping message and an ARP request, respectively.

<sup>3</sup>The reason for this is that only encrypted packets with mathematically weak IVs can be used. The ratio of such IVs to other IVs is small.

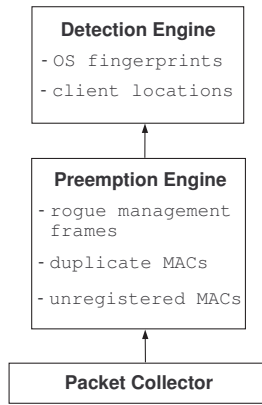


Fig. 1. The software architecture of LIMS.

manner. It is important to consider network performance when making the above decision. On a resource constrained AP, the overall network service could be degraded when all of the components are implemented on it. The details of each component are described below.

#### A. Packet Collector

The packet collector needs to have a network card that runs in promiscuous mode at all times. Its duties are to capture all of the network traffic, and dissect the frames that it receives into IP and TCP components. Thus, information such as client MAC addresses, Service Set Identifier (SSID), channel assignment, encryption status, and beacon interval can be recorded. Moreover, it also filters the collected traffic into user specific streams. The relevant data will be processed by the intrusion preemption and detection engines described in the following subsections.

Some research argues that monitoring a network from a single device, such as an AP, cannot provide comprehensive coverage [20]. Yet, the coverage problem can be solved with the participation of multiple APs or the utilization of standard wireless range extenders. These extenders can be purchased for less than US \$80 from online retailers.

#### B. Intrusion Preemption Engine

The intrusion preemption engine of LIMS is our first line of defense. The basic objective of this component is to prevent activity that could allow an attacker to obtain the secret key.

After obtaining data from the packet collector, the intrusion preemption engine will perform the following actions:

- 1) Unregistered MAC addresses are temporarily stored together with their location information. This is because an attacker might disclose its MAC address to the AP before the knowledge of a legitimate MAC address is acquired. The location information can be obtained by localization schemes proposed in the literature (e.g., [21]). Typically, such localization schemes require 3 to 4 basestations (or APs in our case) with known locations. This requirement is easily satisfied in commodity Wi-Fi networks.

- 2) Duplicate MAC addresses are temporarily removed from the MAC filter so that network access is denied. This can happen when an attacker spoofs a MAC address to that of a

client that is currently connected. The location of any station using this MAC address will be made available by the APs. If one of the locations matches that of a previously unregistered MAC address, the location of the attacker is identified.

- 3) The presence of management frames (e.g., deauthentication frames) will be observed because many active attacks rely on the transmission of forged frames [19]. Although it has been suggested in [22] that management frames in 802.11i be authenticated, the WEP and WPA protocols do not support this functionality. Thus, the intrusion preemption engine needs to keep a record of all management frames that the AP sends out. By doing this, the transmission of a spoofed management frame to a client can be detected, and the AP can choose not to respond to requests from that particular client.

For example, in order to launch a dictionary attack on the shared key used in a WPA-PSK enabled network, an attacker needs to capture the four authentication frames exchanged between a client and the AP. To do this, an attacker may send out a spoofed deauthentication message to a client to force the client to re-authenticate to the AP. In this case, the AP refuses to perform the authentication process with the client. Thus, the attacker is prevented from capturing the frames needed to launch a brute-force attack on the key.

As a complement to the above three tactics, a warning message can be sent to the system administrator whenever a spoofed MAC address or a forged management frame is detected.

The intrusion preemption engine also has the ability to mitigate DoS attacks based on forged management frames. For example, an attacker may periodically transmit spoofed disassociation frames in order to prevent a client from associating with the AP. In cases like this, the AP is instructed to temporarily ignore reassociation requests from the client. The location information of the attacker can also be sent to the system administrator.

Nevertheless, there are some cases where an attacker might go unnoticed by our intrusion preemption system. For example, the attacker might choose to employ the passive listening techniques described in Section III-A. The attacker could also track legitimate MAC addresses for use at a later time. Once the attacker has acquired the secret key, the MAC address of a legitimate but currently not present client can be used. Since these types of activities will go unnoticed by our intrusion preemption engine, LIMS includes the intrusion detection capabilities described in the next subsection.

#### C. Intrusion Detection Engine

Our intrusion detection module uses MAC address information and OS fingerprinting techniques to detect attackers that evade the intrusion preemption engine. The first 3 bytes of a MAC address, the Organizationally Unique Identifier (OUI), allow us to determine the assignee of a particular card company [23]. Therefore, it is possible to link a card's OUI with the OS on the laptop that uses the card. For example, cards with an OUI of 00 – 17 – F2 are known to be used by Apple Inc. in their MacBook line of computers.

Alternatively, information about OS preference can be obtained when users register with the system administrator. A

table mapping each MAC address to the OS preference can be created. These options reflect a tradeoff between potential increased detection accuracy and overall system complexity.

The OS that is actually running on a suspect client can be identified with OS fingerprinting tools. Examples of active and passive OS fingerprinting tools are Nmap [24] and p0f [25], respectively.

With the above information, we can identify potential attackers by looking for inconsistencies between the card manufacturer/preferred OS and the fingerprinting results. A discrepancy may be cause for a “red flag” to be generated about a particular client. We note that a sophisticated attacker can potentially defeat OS fingerprinting tools by modifying the characteristics of the TCP/IP traffic (e.g., ISNs, initial window sizes, and options) that they base their identifications on with a tool such as IP Personality [26]. Scenarios like this may be addressed by using contextual information obtained from users. For example, a profile could be created for each client that indicates their Internet usage characteristics and intrusion detection based on techniques from machine learning and data mining could be used [27].

We also make a note of the following caveat. There are some rare cases where the above OUI-to-OS mapping should not be applied. For instance, a registered MAC address could belong to a PCMCIA card that is used by different laptops. If these computers are running different operating systems, false positives could be generated. Similarly, there are some cases where a card is used by a machine that can boot into multiple operating systems. To reduce the impact of the above scenarios, our mapping based on OS preference can be used.

## V. CONCLUSION

In this paper, we examined a widely accepted myth about passive listening in wireless networks, and provided a detailed description of how to achieve real “passive listening.” We then developed a lightweight intrusion management system called LIMS for commodity Wi-Fi networks. An attractive feature of LIMS is its ability to enhance network security without modifying existing security protocols or using specialized hardware. Specifically, our system utilizes novel techniques to increase network resilience to various attacks that could reveal the WEP/WPA-PSK shared key. The ability to implement LIMS in a plugin manner and the use of free software attest to the practicality of LIMS. Lastly, the LIMS architecture can prevent a certain class of DoS attacks.

As a part of our future work, we plan to deploy LIMS on our department Wi-Fi network. Additionally, we are anticipating the inclusion of new features for LIMS that can further improve its network protection abilities. One such feature is a proactive honeypot that could be used to better preempt various attacks.

## REFERENCES

- [1] S. R. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the key scheduling algorithm of rc4,” in *SAC '01: Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*. London, UK: Springer-Verlag, 2001, pp. 1–24.
- [2] N. Borisov, I. Goldberg, and D. Wagner, “Intercepting mobile communications: the insecurity of 802.11,” in *MobiCom '01*. New York, NY, USA: ACM Press, 2001, pp. 180–189.
- [3] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker, “Security flaws in 802.11 data link protocols,” *Commun. ACM*, vol. 46, no. 5, pp. 35–39, 2003.
- [4] C. He and J. C. Mitchell, “Analysis of the 802.11i 4-way handshake,” in *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*. New York, NY, USA: ACM Press, 2004, pp. 43–50.
- [5] C. He, M. Sundararajan, A. Datta, A. Derek, and J. C. Mitchell, “A modular correctness proof of ieee 802.11i and tls,” in *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2005, pp. 2–15.
- [6] B. Rite, “Cracking wep and wpa wireless networks.” [Online]. Available: [http://docs.lucidinteractive.ca/index.php/Cracking\\_WEP\\_and\\_WPA\\_Wireless\\_Networks](http://docs.lucidinteractive.ca/index.php/Cracking_WEP_and_WPA_Wireless_Networks)
- [7] “Backtrack live cd by remote exploit.” [Online]. Available: <http://www.remote-exploit.org/index.php/BackTrack>
- [8] “Aircrack-ng: an 802.11 wep and wpa-psk keys cracking program.” [Online]. Available: <http://aircrack-ng.org/doku.php>
- [9] Y. Zhang, W. Lee, and Y.-A. Huang, “Intrusion detection techniques for mobile wireless networks,” *Wirel. Netw.*, vol. 9, no. 5, pp. 545–556, 2003.
- [10] A. Adya, P. Bahl, R. Chandra, and L. Qiu, “Architecture and techniques for diagnosing faults in ieee 802.11 infrastructure networks,” in *MobiCom '04*. New York, NY, USA: ACM Press, 2004, pp. 30–44.
- [11] J. Yeo, M. Youssef, and A. Agrawala, “A framework for wireless lan monitoring and its applications,” in *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*. New York, NY, USA: ACM Press, 2004, pp. 70–79.
- [12] M. K. Chirumamilla and B. Ramamurthy, “Agent based intrusion detection and response system for wireless lans,” in *ICC '03. IEEE International Conference on Communications*, 2003, pp. 492–496.
- [13] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, “Characterizing user behavior and network performance in a public wireless lan,” in *SIGMETRICS '02: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM Press, 2002, pp. 195–205.
- [14] D. Schwab and R. Bunt, “Characterising the use of a campus wireless network,” in *INFOCOM: 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, March 7-11, 2004*.
- [15] J. Bellardo and S. Savage, “802.11 denial-of-service attacks: Real vulnerabilities and practical solutions,” in *12th USENIX Security Symposium, Washington D.C., USA, August 2003*. [Online]. Available: <http://www.cs.ucsd.edu/~savage/papers/UsenixSec03.pdf>
- [16] M. Raya, J.-P. Hubaux, and I. Aad, “Domino: a system to detect greedy behavior in ieee 802.11 hotspots,” in *MobiSys '04*. New York, NY, USA: ACM Press, 2004, pp. 84–97.
- [17] P. Kyasanur, “Selfish mac layer misbehavior in wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 5, pp. 502–516, 2005, senior Member-Nitin H. Vaidya.
- [18] “Airdefense enterprise: a wireless intrusion prevention system.” [Online]. Available: <http://www.airdefense.net/>
- [19] P. Mateti, “Hacking techniques in wireless networks.” [Online]. Available: <http://www.cs.wright.edu/~pmateti/InternetSecurity/Lectures/WirelessHacks/Mateti-WirelessHacks.htm>
- [20] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, “Enhancing the security of corporate wi-fi networks using dair,” in *MobiSys '06*. New York, NY, USA: ACM Press, 2006, pp. 1–14.
- [21] M. P. F. Koushanfar, S. Slijepcevic and A. Sangiovanni-Vincentelli, “Location discovery in ad-hoc wireless sensor networks,” *Ad Hoc Wireless Networking*, (editors X. Cheng, X. Huang and D.-Z. Du).
- [22] C. He and J. C. Mitchell, “Security analysis and improvements for ieee 802.11i.” in *NDSS*, 2005.
- [23] “Ieee ra frequently asked questions.” [Online]. Available: <http://standards.ieee.org/faqs/OUL.html>
- [24] “Nmap.” [Online]. Available: <http://insecure.org/nmap/>
- [25] “p0f: a versatile passive os fingerprinting tool.” [Online]. Available: <http://lcamtuf.coredump.cx/p0f.shtml>
- [26] “Ip personality: a netfilter module to change characteristics of network traffic.” [Online]. Available: <http://ippersonality.sourceforge.net/>
- [27] M. A. Maloof, *Machine Learning and Data Mining for Computer Security: Methods and Applications (Advanced Information and Knowledge Processing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.