A Hybrid Rogue Access Point Protection Framework for Commodity Wi-Fi Networks

Liran Ma, **Amin Y. Teymorian**, **Xiuzhen Cheng**
Department of Computer Science
The George Washington University,
Washington DC, 20052, USA.
Email: {lrma,amin,cheng}@gwu.edu

***Abstract*—We develop a practical and comprehensive hybrid rogue access point (AP) detection framework for commodity Wi-Fi networks. It is the first scheme that combines the distributed wireless media surveillance and the centralized wired end socket level traffic "fingerprinting." The former is designed not only to detect various types of rogue APs, but also to discover suspicious activities so as to prevent the adversaries from turning victim APs into rogue devices. Moreover, the socket level traffic fingerprinting helps our frame work to achieve a finer granularity on rogue AP detection among the existing schemes. This framework has the following nice properties: i) it requires neither specialized hardware nor modification to existing standards; ii) the proposed mechanism greatly improves the rogue AP detection probability so that network resilience is improved; iii) it provides a cost-effective solution to Wi-Fi network security enhancement by incorporating free but mature software tools; iv) it can protect the network from adversaries capable of using customized equipment and/or violating the IEEE 802.11 standard; v) its open architecture allows extra features to be easily added on in the future. Our analysis and evaluation demonstrate that this hybrid rogue AP protection framework is capable of reliably revealing rogue devices and preempting potential attacks.**

***Index Terms*—Rogue access point detection, commodity Wi-Fi networks, intrusion detection, wireless security.**

I. Introduction

With the increasing popularity of Wi-Fi networks, securing such a network becomes a challenging problem. Commodity Wi-Fi networks are particularly vulnerable to attacks because of factors such as open medium, insufficient software implementations, potential for hardware deficits, and improper configurations. Among all the security threats, one of the most dangerous hazards is the prevalence of rogue APs. A rogue AP is typically referred to as an *unauthorized AP* in the literature. This type of device can be easily deployed by end-users. When a rogue AP is connected to a network, it can be used by adversaries for committing espionage and launching attacks.

Similarly, *improperly configured APs* and *phishing APs* can introduce the same security threats once exploited by adversaries. Therefore, they can be regarded as rogue APs as well. More importantly, there is a more insidious type of rogue APs, called the *compromised APs*, that has drawn little attention in the literature. A compromised AP is the most dangerous rogue AP that can exist in commodity Wi-Fi Networks. In particular, it is difficult to detect such a rogue device because the AP itself is not malfunctioning (e.g., operating without specified security controls). Further, the AP does not display anomalous misbehavior such as broadcasting a duplicate SSID. Thus, a compromised AP can significantly diminish the overall security of the network. A summary of the types of rogue APs and a number of possible scenarios is shown in Table I. For a detailed taxonomy of rogue APs, we refer the readers to Ref. [1].

Rogue AP Class	Possible Scenarios

1. Improperly configured	insufficient security knowledge;
	faulty driver; physically defective;
	multiple network cards

2. Unauthorized	connected to internal LAN without permission; external neighborhood AP
3. Phishing	fabricated by adversary
4. Compromised	disclosure of security credentials

TABLE I
Rogue AP Taxonomy and Scenarios.

According to an early study by Gartner [2], rogue APs are present on about 20% of all enterprise networks. The main reason leading to this phenomenon is that advances in hardware and software have made AP installation, AP discovery (e.g., finding improperly configured APs), and AP compromise an easy task for attackers. It is convenient to obtain an AP and plug into a network without being discovered for some time. Moreover, commodity Wi-Fi network cards that have the capability to capture all 802.11 transmissions can currently be purchased for about US $30 on eBay. Hence, the process of driving around and looking for vulnerable APs (known as "wardriving") can be accomplished by people with limited security backgrounds. In addition, the probability that an unprotected AP can be exploited is increased by people called warchalkers that document and publicize the locations of APs.

To make matters worse, a properly configured AP with security features enforced can still be compromised, thus becoming a rogue AP. As shown in [3]–[5], the most common security protocol, Wired Equivalent Privacy (WEP), has been shown to be breakable even when correctly configured. Recently, Wi-Fi Protected Access (WPA) has been created in response to the serious weaknesses that researchers found in WEP. However,

978-1-4244-2026-1/08/$25.00 © 2008 IEEE

WPA does not necessarily work with the first generation APs. 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. Another deficiency of WPA is that it still relies on the RC4 encryption algorithm. Due to these weaknesses in WEP and WPA, an attacker can easily compromise an AP and turn it into a rogue one.

Facing such unprecedent challenges, the traditional way of protecting networks with encryption and firewalls is no longer sufficient. Thus, several techniques are proposed to detect the existence of rogue APs in literature. One of the most popular approaches is to scan the area of interest with a wireless device running on laptops or handheld devices. This idea is also widely adopted in the commercial products with advanced features such as non-interactive scanning and continuous monitoring capabilities enabled. However, there still lacks a satisfactory and practical solution that is competent enough to tackle rogue APs.

We develop a novel hybrid framework for protecting Wi-Fi networks from rogue APs. In this framework, rogue APs are automatically detected and located through the combination of a distributed wireless scanning and a centralized traffic "fingerprinting." Accordingly, it includes two major components: a distribution detection module (DDM) and a centralized detection module (CDM). The former can be connected to or implemented on APs as small plugins, while the latter is located at the gateway router of a local network. In addition, our framework works in conjunction with current security protocols such as WEP and WPA, and it does not require any specialized wireless hardware. Furthermore, it can protect the network from adversaries using customized equipment and/or violating the IEEE 802.11 standard. Lastly, it works consistently under various network configurations.

The rest of the paper is organized as follows. Section II discusses related work. The proposed rogue AP protection framework is elaborated in Section III, and the detailed analysis is provided in Section IV. The evaluation results are presented in Seciont V. Finally, our conclusion and future research directions appear in Section VI.

II. Related Work

Due to the security threats that a rogue AP can pose for corporate Wi-Fi networks, detecting such APs is one of the most important tasks of an IT department. Traditional rogue AP detection relies on network enumeration tools (e.g., NetStumbler) running on laptops or handheld devices carried by IT personnel. This "walking audit" approach is both time-consuming and unreliable. Further it fails when a rogue AP spoofs characteristics such as the MAC address and Service Set Identifier (SSID) of a legitimate AP.

To help automate the scanning process and provide continuous monitoring capabilities, a number of commercial products have been developed [6]–[8]. AirDefense [6] is one such product. It uses a combination of radio frequency sensors and an intrusion detection/protection server appliance to capture, process, and correlate network events. However, the latest release, AirDefense 7.2, has a starting price of US $\$7,995$. Lastly, if the specialized monitoring sensors are not used, it is difficult to guarantee a complete coverage of the network to ensure effective rogue AP detection.

On the other hand, the research community has just recently started to direct attention toward rogue AP detection. An architecture for fault diagnostics in IEEE 802.11 networks is presented in [9]. Multiple APs and mobile clients perform RF monitoring to help detect the presence of rogue wireless devices such as unauthorized APs. Each client is required to install special diagnostic software, and rogue APs are assumed to transmit beacon messages and respond to probe requests. In contrast, our framework does not inconvenience clients with additional software installs. Further, its detection ability is not based on the assumption that rogue APs will function properly.

Bahl *et al.* [10] propose a distributed monitoring infrastructure called DAIR. It attaches USB wireless adapters to desktop machines for more comprehensive traffic capturing ability. Although techniques to reduce false positives/negatives are provided, its effectiveness is still dependent on AP functionality that can be easily turned off. Additionally, both of [9] and [10] assume that some specific characteristics of IEEE 802.11 standards cannot be violated by the adversaries. Conversely, our proposed framework avoids their limiting dependencies, and provides protection from types of rogue APs that they cannot detect.

Differences in inter-packet spacing between traffic flows on wired and wireless networks is used in [11], [12] for identification of rogue APs. However, the scheme does not differentiate between wireless traffic from authorized and unauthorized APs. It also assumes that APs will be connected within one hop to a switch monitoring the traffic, and relies on visual inspection of traffic characteristics.

Multiple network sniffers are used in [13] for detecting rogue APs and eavesdroppers. Each sniffer has three network cards, and the intrusion detection capabilities are stymied by MAC address spoofing. Yeo *et al.* [14] 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. Our framework provides techniques to detect rogue APs that have spoofed MAC addresses without relying on heavily equipped sniffers. It can also detect sophisticated eavesdroppers and avert AP compromise.

Recently, two passive online rogue AP detection algorithms are proposed in [15]. The core of these two algorithms are the sequential hypothesis tests applied to packet-header data that are passively collected at a monitoring point. Both algorithms exploit the fundamental properties of the 802.11 CSMA/CA mechanism and the half duplex nature of wireless channels to differentiate wired and wireless LAN TCP traffic. Once TCP ACK-pairs are observed, prompt decisions are made with little computation and storage overhead. Yin *et al.* [16] propose a layer-3 rogue AP detection approach using the combination of a verifier and wireless sniffers. In this approach, a verifier on the internal wired network is employed to send test traffic towards wireless edge. Once wireless sniffers capture an AP relaying the test packets, the AP is flagged as rogue. In addition, binary hypothesis testing technique is adopted to improve the robustness of detection.

Our proposed framework differs from previous work in that it provides robust and comprehensive protection against rogue APs through a novel coupling of rogue AP preemption and detection. It also defends against a more insidious type of rogue APs, i.e., the compromised APs, that has never been addressed in the literature before. Further, it can detect rogue APs that have the ability to violate the IEEE 802.11 standard. Moreover, the mature techniques and freely available software that this framework employs make it an efficient and cost-effective solution. Lastly, modifications to the underlying wireless standard are not necessary with this framework.

III. The Framework Design

This hybrid framework is designed to monitor network activities, forestall events that could lead to the generation of rogue APs, discover existing rogue APs, and block unauthorized network access through rogue APs. The two main components that constitute its architecture are the distributed monitoring module and the centralized detection module. The former should be placed in a way that it can cover the interested areas as much as possible. On the other hand, the latter is located at the gateway router of a local network, which allows it to examine all the traffic coming in and going out of the network. A brief summary of each component and its functionalities is listed in Table II.

The type of Wi-Fi networks 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. Rekeying (i.e., change of the WEP secret key) is not a desirable action, as it incurs significant overhead. The layout of an example network can be found in Fig. 2, where a wireless network is behind a NAT box.

A. The Distributed Detection Module

This module consists of a passive wireless frame collector, a rogue AP preemption engine, and a rogue AP detection engine. An illustration of the overall architecture of the distributed monitoring module can be seen in Fig. 1. The frame collector is responsible for gathering wireless traffic. The collected data is then passed to the preemption engine, where checks are performed in order to thwart various attacks. Finally, the data is analyzed by the detection engine. There are also probing functions shared by the preemption and detection engines so that adversaries can be lured into revealing their presence.

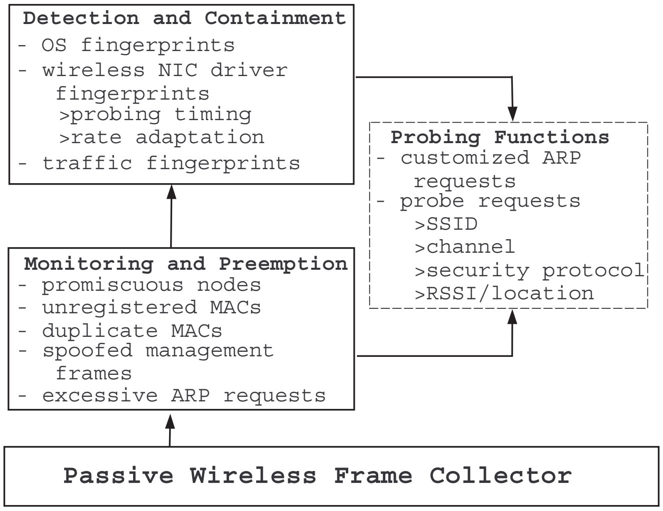


Fig. 1. Distributed monitoring module architecture.

1) Passive Wireless Frame Collector: A frame collector is needed for realtime WLAN monitoring so that rogue wireless devices can be quickly identified, and network administrators notified when appropriate. One benefit of the frame collector is its natural ability to separate wired and wireless traffic. Thus, there is no need for complicated modules that attempt to isolate the two by examining traffic signatures.

The frame collector needs to have a network device that runs in promiscuous mode at all times. The entire region of interest can be covered with the assistance of wireless range extenders. One of the frame collector's main duties is to capture all wireless traffic. Subsequently, the frame collector dissects frames into IP and TCP components. This allows for information such as client MAC addresses, SSID, channel assignment, encryption status, and beacon interval to be recorded. It also filters the collected traffic into user specific streams such as AP-client pairs. The relevant data will be processed by the intrusion preemption and detection engines described in the following subsections.

Note that it is critical to hide the wireless frame collector described from adversaries. Otherwise, a prudent attacker may change tactics to elude capture. Techniques that can achieve real "passive listening" are detailed in [17].

2) Preemption Engine: While attempted network attacks cannot be avoided, it is possible to prevent some attacks before they happen. In particular, a certain amount of information must be collected by an adversary before an attack can actually occur. The prompt identification of such activity can help thwart an impending attack. Subsequently, a rogue AP preemption engine is included in the proposed framework.

The rogue AP preemption engine is our first line of defense. The basic objectives of this component are to trap sniffers and thwart activity that can lead to AP compromise. Probing of potential eavesdroppers and network integrity checks are performed to accomplish these goals. The former is designed to discover passive listeners while the latter is used to prevent

TABLE II
FRAMEWORK COMPONENT SUMMARY

Component		**Security Threat**	**Countermeasure**
Distributed Detection Module	Wireless Frame Collector	Rogue wireless traffic (e.g., unauthorized access, passive attacks, and active attacks)	Device running in promiscuous at all times with "real passive" listening capability; off-the-shelf wireless range extenders
	Preemption Engine	Class 4 rogue APs; passive listeners; network integrity violations (e.g., unregistered MACs, duplicate MACs, and forged management frames)	Probing eavesdropper (e.g., using ARP requests); logging and localization of unregistered or duplicate MACs; avoid post-forge requests
	Detection Engine	Class 1-4 rogue APs; attackers that evade Preemption Engine or obtain the security credentials	Active AP probing; location verification; client disassociation; passive or active OS fingerprinting
Centralized Detection Module	Scanning & Filtration	Rogues undetected or uncovered by DDM (e.g., rogues hide behind NAT devices)	Port 80 scans; socket-level stream analysis (e.g., RRTs or inter-packet arrival times)

a legitimate AP from being compromised.

a) Eavesdropper Probing: Probing functionality is employed to help prevent Class 4 rogues from appearing on a network. In particular, messages are periodically generated that, when replied to, reveal the presence of a sniffer. One type of message is an ARP request. If a potential attacker is eavesdropping at the network traffic and replies to one of the trap ARP requests, her presence is revealed.

Since these broadcasted messages are regular network signaling traffic, it is unlikely that an attacker will notice the existence of our preemption engine. In addition, the interval selected for broadcasting the frames reflects a trade off between available bandwidth consumption and time needed for detection. These parameters can be customized based on the capabilities of the underlying hardware systems.

b) Intruder Discovery: After obtaining data from the wireless frame collector, four integrity checks outlined below will be carried out. We assume that every legitimate wireless device's MAC address is available to the preemption engine.

- Unregistered MAC addresses are temporarily stored together with their possible 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. Additionally, these MAC address will be shared with the rogue AP detection engine described in Section III-A3.
- 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 presence of management frames (e.g., deauthentication frames) will be observed and checked because many active attacks rely on the transmission of forged frames [18]. Once it is determined to be a spoofed frame, APs refuse to respond to that frame.
- The appearance of excessive ARP requests in a given time is the sign of a *ARP request replay attack* [19], which repeatedly transmits the same ARP packet to obtain new *initial vectors* (IVs) from an AP in order to crack a WEP key. In cases like this, an AP is instructed to temporarily ignore such ARP requests.

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

3) Detection Engine: There are two primary reasons for the rogue AP detection engine. First, defending against Class $1-3$ rogue APs is an inherently reactive process. For example, there is no way to prevent an attacker from setting up a phishing AP outside of a private organization. The AP probing technique described in Section III-A3a is used to lure Classes $1 - 3$ rogue APs into revealing their presence. Secondly, a sophisticated adversary may be able to evade the preemption techniques for Class 4 rogue APs. Class 4 rogue APs are detected by first identifying traffic from an unauthorized user. Additional mechanisms are included for handling adversaries that are strong enough to use hardware that violates the 802.11 standard.

We assume that a floor plan of the building containing the to-be-protected network is available. In particular, the exact location of authorized APs and range extenders should be known. The above location data, along with information such as an AP's MAC address, SSID, nearest extender, working channel, and typical received signal strength indicator (RSSI) can be made accessible to the detection engine.

a) AP Probing: An AP advertises its presence several times per second by broadcasting special frames called beacons that carry its SSID. Stations can discover an AP by passively listening for beacons, or by transmitting a probe request message to actively search for an AP with a specified SSID. Our detection engine uses active honeypot functionality[1] to discover rogue APs by sending out probe requests. It is capable of detecting the first three classes of rogue APs.

Therefore, a particular AP can be discovered from its probe responses. The next step is to determine whether or not it is a rogue AP. One way to do this is to compare the discovered APs with those belonging to a list of authorized APs. Any AP that is detected and does not appear in the authorization list can

[1] Examples of active honeypot systems include Strider HoneyMonkeys [20] and the Honeyclient Project [21].

be labeled as a rogue device. The relevant values associated with each AP in the table of authorized APs include its MAC address, SSID, working channel, and equipment vendor.

Accordingly, our detection system has a probe request frame periodically sent out on all of the channels (e.g., 11 channels in 802.11b). This property increases the likelihood of a rogue AP being detected because any AP that hears the request will send a probe response back to the detection engine. In this response, information such as the MAC address must be included, even though the SSID may not be present. If the reported MAC address matches an unregistered MAC address found during an integrity check, we can conclude that it belongs to a rogue AP. Finally, the switch port that is associated with the rogue AP's MAC address can be closed to eliminate it from the network.

In the event that a rogue AP spoofs a legitimate AP's MAC address and SSID, location information should be used to make a judgement. If an AP announces a legitimate MAC address, but has localization results that are inconsistent with those in the AP MAC-to-location table, it can be considered to be a rogue AP.

b) Compromised AP Detection: A compromised AP is detected by identifying an unauthorized client who is connecting to it. The client can be detected by employing a combination of various fingerprinting techniques. Fingerprinting is a process by which a device or the software it is running is identified by its externally observable characteristics. We detail these techniques below.

i) User-specific traffic fingerprinting: A profile could be created (either online or offline) for each client that indicates their network traffic patterns such as web browsing preferences. More specifically, the profile contains distinctive identifiers such as Internet destinations and email servers. In this case, techniques from machine learning and data mining [22] could be performed.

ii) OS fingerprinting: The OS that is running on a suspect client can be identified with OS fingerprinting tools. Examples of active and passive OS fingerprinting tools are Nmap [23] and p0f [24], respectively. Information about OS preference can be obtained when users register with the system administrator. With this information, we can identify potential attackers by looking for inconsistencies between the fingerprinting results and the preferred OS. A discrepancy may be cause for a "red flag" to be generated about a particular client.

iii) Wireless network interface card (NIC) driver fingerprinting: Due to the ambiguity of the 802.11 standard, different implementations of the same protocol specification in wireless device drivers behave differently. Fingerprinting techniques take advantage of these implementation-dependent differences to accurately identify a driver. We list two implementation-dependent algorithms below.

- *Timing of probe request frames*: The algorithm used to scan for AP is not explicitly defined in the 802.11 standard. Therefore, it has led to the development of many wireless device drivers that display different characteristics in terms of this function [25]. Using these characteristics, wireless drivers can be determined with acceptable accuracy.
- *Rate adaptation schemes*: The lack of an explicit specification for a rate adaptation algorithm results in different implementations such as ARF [26] and SampleRate [27]. Thus, each algorithm will have a different impact on observable traffic characteristics such as throughput and occurrence of retransmissions. Techniques have been proposed in [28] to distinguish different drivers by rate adaptation algorithms.

iv) Client location fingerprinting: The ability to distinguish the location of spoofed nodes from the authentic nodes can help to identify compromised APs. A commonly used location distinction approach is RSSI measurement. Yet, RSSI values may vary due to small-scale and frequency-selective fading. There are some more advanced techniques proposed for location distinction such as *link temporal signature* [29].

B. Centralized Detection Module

The wireless distributed monitoring module is effective at spotting rogues, but those not within the surveillance coverage range may run away undetected. In addition, it is conceivable that there might be no wireless network in a company. As a result, there is no AP or wireless extender available. Therefore, the ideal method of detecting such rogue APs is to use a central console attached to the wired side of the network for monitoring. Another benefits of a central point detection is that it alleviates the need to walk through the facilities in case of incomplete coverage. It can be regarded as a compensate to the wireless monitoring module. In the following, we elaborate some novel techniques for finding potential rogue APs from the wired side of the network.

1) Port 80 *Detection:* Most commodity AP products enable port 80 (HTTP) in order to let the management personnel to login. Thus, a simple but efficient method to detect the existence of an rogue AP is to trap it into responding to queries. In our centralized detection module, a port 80 scanning program is running on the console that is connected to the gateway router. The port scan program can identify enabled TCP ports from various devices connected to the wired network. To be specific, it uncovers all port 80 (HTTP) interfaces on the network, which includes all Web servers, and all APs. Even if an rogue AP's port 80 interface is disabled or protected by a username and password, it will generally respond to the port scanner's ping with the vendor name and its corresponding IP address. After obtaining these information, they are compared with the stored data of the authorized APs. Any mismatching will trigger a security alert to the system administrator. With the IP address of a suspected AP, it is possible to determine its physical location via router table entries.

There are two extra advantages to carry out the port 80 scan from the wired side of the local networks compared to the wireless side. Firstly, there is much more available bandwidth than the wireless links. Secondly, it is more "passive" than sending out queries via wireless broadcasting because the query traffic is more invisible inside wired communications. One example of such port scanning tool is SuperScan [30], which is free available from Internet.

2) Socket Level Wireless Traffic Detection: The port 80 scan will fail if a rogue AP does not have port 80 HTTP service enabled. Consequently, it is necessary to scrutinize at a finer granularity so as to discover the rogue APs. A fairly effective approach is to conduct socket level TCP/UDP traffic inspection. It is important to notice that we adopt socket level traffic examination instead of the popular IP address based method. Since we consider the wireless networks that are behind network address translation (NAT) boxes, hosts may use either Ethernet or WLAN to connect to a NAT box. All traffic through a NAT box will have the IP and MAC address of the NAT box. As a result, the traffic features reflected in the IP level are no longer wireless unique, which makes the IP address scanning based schemes [11], [15] insufficient. Thus, it is necessary to pin down the wireless links via the socket level traffic inspection.

The socket level inspection is grounded on the concept that wireless links uses a contention based MAC protocol to access the shared link, which naturally causes a longer random delay. Therefore, wireless links cause more random temporally different spreading of packets as compared to wired links. Parameters such as packet inter-arrival (or inter-departure) time, round trip time (RTT), and etc., can be employed to identify wireless links as what is done at the IP level. In addition, some of the above features are also applicable for UDP traffic.

Here, we adopt the inter-packet spacing, which is the spreading of packets, to differentiate wireless links from wired links. In general, the inter-packet of a wireless link is greater than that of a wired link. To be specific, when two back-to-back packets are sent on a perfect wireless channel, the inter-departure time of the packet pair is uniformly distributed between 500 μs and 1130 μs, with a median of 810 μs [15]. Although an Ethernet connection uses shared media, the randomness caused by the shared media in Ethernet is negligible compared to the one in a wireless network because of its high bandwidth and ability to detect collisions. Thus, a link is labeled as wireless if its corresponding socket level traffic shows the above pattern in packet inter-departure time.

IV. Analysis

In practice, most of the Wi-Fi networks are behind a NAT box due to the lack of IPv4 addresses. As a result, all traffic that go through a NAT box have the same IP/MAC address (i.e., the IP/MAC address of the NAT box). Considering these characteristics, we illustrate an example Wi-Fi network that is protected by our framework in Fig. 2. The detailed analysis of the proposed framework based on this example network is described in the following subsections.

A. Coverage of DDM

Some research argues that monitoring a network from devices such as APs cannot provide comprehensive coverage [10]. Yet, the coverage can be greatly improved with the participation of multiple APs (e.g., AP *a* and *b* in Fig. 2) and the utilization of standard wireless range extenders (e.g., ranger extender *a* and *b* in Fig. 2). These extenders can currently be purchased for less than US $80 from online retailers. Based on the specifications of an off-the-shelf extender (e.g., the Belkin F5D7132), it can scan all working channels of 802.11b and 802.11g with a working range up to 457.2 meters.

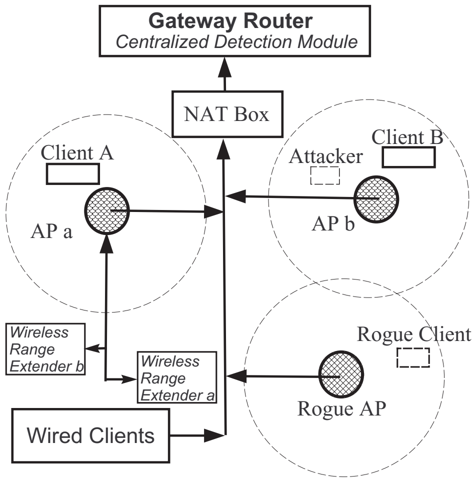


Fig. 2. A typical Wi-Fi network scenario: AP a & b, and wireless range extender a & b compose the Distributed Detection Module.

Still, it is possible that neither APs nor extenders can pickup the wireless signals sent by a customized wireless antenna that can limit its signal inside a very small area. Subsequently, the task of identifying a wireless link falls upon the centralized wired end detection module (located at the gateway router in Fig. 2).

B. Attack Prevention by DDM

For example, in order to launch a dictionary attack on the shared key used in a WEP or a WPA-PSK enabled network, an attacker in Fig. 2 needs to capture the four authentication frames exchanged between the client *B* and the AP *b*. To do this, the attacker needs to send out a spoofed deauthentication message to client *B* to force the client to re-authenticate to the AP. After capturing authentication frames, the hash of each word in a dictionary is compared to the hashed passphrase used during the handshake. Of course, if the passphrase is strong enough (i.e., not located in the dictionary), this attack fails.

In this case, the preemption engine of DDM will detect the fake deauthentication message and instruct the AP to refuse performing 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.

By preempting intruders that could reveal the secret network key, we prevent the creation of Class 4 rogue APs. Nevertheless, there are some cases where an attacker might get away unnoticed by our preemption system. For example, the attacker might choose to employ the passive listening techniques described in [17]. 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 may escape from our checking functionalities, the rogue AP detection capabilities is introduced in the next subsection.

C. Rogue Device Discovery

1) Why Does the AP Probing Work?: There is a common misconception that disabling the “Broadcast SSID” option in an AP hides the SSID. As a result, a rogue AP is able to escape from AP probing. In reality, disabling this feature only makes the AP transmit a null (zero-length) SSID in beacon frames and probe responses instead of the actual SSID. There are still several other frames (e.g., probe requests, association requests, and reassociation requests) that carry the SSID. Hence, it is impossible to keep an SSID value secret without manually reconfiguring device drivers or hardware to violate the 802.11 standard.

In addition to regular rogue device, AP probing also handles extreme cases where rogue APs have had their driver and/or firmware modified in such a way that neither beacon frames nor probe response frames are transmitted. Therefore, there is no MAC address information available to draw a conclusion. Nevertheless, a disassociation message can be sent from AP *b* to the rogue client of the rogue AP in Fig. 2 based on the information collected by the wireless frame collector. When the rogue client sends out a reassociation request, the MAC address and SSID of the rogue AP will be disclosed.

Note that the above technique can thwart an adversary with a level of strength that has never been assumed before. In particular, other work such as [9] and [10] assume that an attacker does not have the ability to violate characteristics of the 802.11 standard. Although this assumption is reasonable in many cases, the protection of any system based on it can be undermined. Our framework does not place this limitation on the capabilities of the adversary. Hence, it is able to provide both robust and comprehensive protection from rogue APs.

2) Pitfalls in Compromised AP Detection Engine: We make a note of the following caveat in the compromised AP detection unit proposed in Subsection III-A3b. There are some rare cases where the proposed fingerprinting techniques 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. It is also possible for a sophisticated attacker to defeat fingerprinting tools by modifying the characteristics of the TCP/IP traffic (e.g., ISNs, initial window sizes, and options) and driver behaviors that they base their identifications on. A freely available tool that performs such functions is IP Personality [31].

D. Remarks

This framework is a marriage of the distributed wireless media monitoring module and the centralized wireless rogue equipment detection module. The former module plays a more important role in the framework in that it identifies rogue APs and preempts possible attacks via directly measuring ongoing wireless communications. The latter works as a necessary compliment to the former.

In these two modules, several novel techniques are introduced to detect and prevent the existence of rogue APs. A complete classification of these methods and the security threats they mitigate are listed in Table II. Although some of the techniques have been vetted in the literature, the elegance and comprehensiveness of such a hybrid approach to rogue AP protection have never been achieved before. Moreover, powered by the idea of pining down to socket level traffic feature recognition, our framework achieves the highest rogue detection granularity among the existing schemes. Lastly, it has been shown in the above analysis that the proposed framework is capable of discovering rogue devices with a high probability and a low overhead.

V. Empirical Evaluation

The focus of our evaluation is on the detection and prevention of compromised APs because they are the most insidious rogue APs. We follow two approaches. The first uses network traces gathered from multiple monitoring points at the 2004 SIGCOMM conference [32] to examine compromised AP detection techniques (e.g., the fingerprints of wireless clients). The second employs controlled real experiments to evaluate the preemption engine.

A. Trace-Based Study

The widely used SIGCOMM conference trace [32] was collected using standard wireless cards in monitor mode with tcpdump-like tools for a span of 5 days. For privacy concerns, IP and MAC addresses are anonymized consistently throughout the entire trace (i.e., there is a unique one-to-one mapping between addresses and anonymous “marks”). Thus, each user is uniquely identified by its anonymous mark.

Using the aforementioned trace, we evaluate the user-specific traffic fingerprinting techniques proposed in Subsection III-A3b. The preferred OS, rate adaptation scheme, and user location are not evaluated due to the lack of a ground truth. A number of training samples are requited to build an Internet traffic (e.g., <IP address, port number>) profile for a particular user. The training period may vary depending on the networking activities of the user. According to Ref. [33], a one day span of training samples is enough to profile a large percentage (70%-100%) of users in this trace. Therefore, the remainder of the trace (about 4 days) is used as validation data in our study. [2] To simulate the effect of a compromised AP with an unauthorized user connected to it that spoofs the

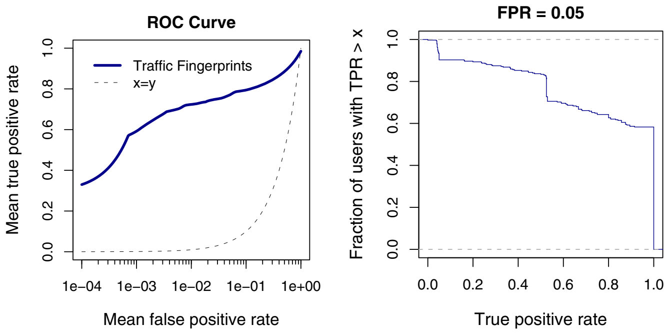
[2] We exclude users that are not present in either the training or the validation samples.

identity of a legitimate user A, the validation samples from users other than A are classified based on A's profile using the Bayesian approach [34].

The two metrics that we employ to evaluate the performance of our proposed fingerprinting techniques are as follows:

- The true positive rate (TPR) refers to the percentage of validation data that user A does not generate but we correctly classify as unauthorized users.
- The false positive rate (FPR) refers to the percentage of validation data that user A generates but we incorrectly classify as unauthorized users.

We detail our evaluation results below. The relation between the mean TPR and the mean FPR, also known as the receiver operating characteristic (ROC) curve, is plotted in Fig 3(a). For reference, the dotted line $x = y$ represents the performance of random guessing. The proposed techniques clearly achieve a high accuracy in terms of unauthorized user detection on average. For example, the mean TPR reaches over 80% when the mean FPR is 0.05. This high average accuracy is partly due to the rich diversity that lies in the attendees of the conference. A large portion of the attendees come from different universities, and therefore become distinguishable when accessing their school web and email servers. Additionally, the complementary cumulative distribution function (CCDF) of the TPR when the FPR is fixed to be 0.05 is shown in Fig 3(b). We see that over 60% unauthorized users can be identified at a TPR that is at least 95% on average.



(a) Mean TPR vs. FPR. (b) The CCDF of TPR.

Fig. 3. TPR and FPR Measurements.

B. Experimental Study

Since link-layer encryption is not employed by the SIGCOMM network, there is no attack targeting AP compromise that can be used to evaluation our proposed preemption techniques. Therefore, we used controlled real experiments conducted in a real campus setting. We describe our experiment setup, methodology, and results in the subsections that follow.

1) Setup: The prototype rogue AP preemption engine is on a Dell desktop (Intel Pentium IV 3.2GHz, with 1.5 GBytes of memory) running Linux 2.6.20, and equipped with two wireless interfaces. One is an external Trendnet TEW-443PI PCI card (Atheros AR2414a chipset) using MadWifi [35] driver. Enabled by the MadWifi driver, this interface functions as a normal AP. The other interface is an external Linksys WMP54G PCI card, which serves as the host for all the rogue AP detection modules including the preemption engine. The prototype preemption engine employs open source programs such as Kismet [36] and Nmap [23].

The attacker is a Dell laptop with a number of wireless surveillance and hacking tools such as Kismet and Aircrack [37], respectively, installed. Therefore, this laptop is capable of launching various attacks that include management frame forgery (e.g., deauthentication flooding, and fake authentication), *ARP request injection* [19], *fragmentation* attack [37], and *Monkey-Jack* attack [38]. We arrange the attacker to launch the above attacks at various times and locations.

2) Methodology: Experiments are done during weekends when the offices are empty in order to minimize the impact of external factors such as traffic from other APs or client devices. Our setup completely separates normal AP functionalities from rogue AP detection modules. As a result, it is easy for us to debug. The machines are tested to ensure that packet collection (i.e., "logging") and processing does not degrade the performance of the wireless network or inadvertently impact the results of the experiment. We have more control over the system compared to commodity AP products because a software programmable AP is constructed using widely available devices that are reasonably priced. We also adopt open source programs to greatly reduce development costs.

3) Results: Our logged records show that the preemption engine reliably detects all the arranged attacking activities (i.e., it has 100% recall). To be specific, each attack is identified and recorded in log files, which matches with its scheduled time and date. However, our preemption engine generates a few false positives in the presence of external users. This phenomenon occurs when an external client device tries to probe and connect to our AP, even without malicious intent. Therefore, it may be mistakenly labeled as suspicious. Since the external client does not have the necessary credentials to associate with our AP, it repeatedly probes/querys the AP. These activities are indicators of wireless reconnaissance, which is an essential step that precedes many attacks.

In order to measure the storage overhead of the data collected in the distributed detection module, we conduct a test using the Wi-Fi network (802.11 b/g compatible) of the Department of Computer Science at The George Washington University. There are about 20 to 30 users active each day. A monitor machine is placed near an AP to capture wireless frames. The capture process lasts for two weeks. The average data collected per day per AP is approximately 1 Gigabyte. Therefore, by recycling the collected data every week, the storage overhead can be limited to about 7 Gigabyte per AP. This is a reasonable overhead for even low-end computing equipment.

VI. Conclusion

In this paper, we develop a practical hybrid framework targeting preempting attacks that can create rogue APs, and detecting the presence of such devices when they exist. It is the first framework that correlates alerts containing all data from both wired scans and wireless surveillance. An attractive feature of the proposed framework is that it requires neither specialized hardware nor modification to existing security standards. Further, it can be connected to or implemented on APs as small plugins. It also makes use of freely available mature software in order to provide a cost-effective security solution. Lastly, it can protect networks from rogue APs even when assuming that adversaries have the ability to use customized equipment that violates the IEEE 802.11 standard. Our framework is the first one that can successfully protect the network under that assumption.

As a part of our future work, we plan to evaluate the proposed framework in a more open environment, where there are more background "noises" that may cause false positives. Additionally, we are anticipating the inclusion of new features for the framework that can further improve its network protection abilities. One such feature is a proactive honeypot function that can be used to better preempt various attacks.

Acknowledgment

This research was supported by the US National Science Foundation under grant CCF-0627322.

References

[1] L. Ma, A. Y. Teymorian, X. Cheng, and M. Song, "RAP: Protecting commodity wi-fi networks from rogue access points," in *QShine '07: Proceedings of the 4th international conference on Quality of service in heterogeneous wired/wireless networks*, 2007.

[2] "Gartner advises on security." [Online]. Available: http://www.gartner.com/5_about/press_releases/2001/pr20010809b.html

[3] 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.

[4] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," in *MobiCom '01*, 2001, pp. 180–189.

[5] 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.

[6] "AirDefense enterprise: a wireless intrusion prevention system." [Online]. Available: http://www.airdefense.net/

[7] "AirMagnet: Enterprise WLAN management." [Online]. Available: http://www.airmagnet.com/

[8] "Airwave: Wireless network management." [Online]. Available: http://www.airwave.com/

[9] A. Adya, P. Bahl, R. Chandra, and L. Qiu, "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks," in *MobiCom '04*, 2004, pp. 30–44.

[10] 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 2006: Proceedings of the 4th international conference on Mobile systems, applications and services*. New York, NY, USA: ACM Press, 2006, pp. 1–14.

[11] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland, "Rogue access point detection using temporal traffic characteristics," in *GLOBECOM*, 2004.

[12] S. Shetty, M. Song, and L. Ma, "Rogue access point detection by analyzing network traffic characteristics," in *MILCOM*, Orlando, Florida, October 2007.

[13] 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.

[14] 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.

[15] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. Towsley, "Passive online rogue access point detection using sequential hypothesis testing with tcp ack-pairs," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2007, pp. 365–378.

[16] H. Yin, G. Chen, and J. wang, "Detecting protected layer-3 rogue APs," in *IEEE BROADNETS '07: Fourth Annual International Conference on Broadband Networks*, 2007.

[17] L. Ma, A. Y. Teymorian, and X. Cheng, "Passive listening and intrusion management in commodity wi-fi networks," in *GLOBECOM*, 2007.

[18] P. Mateti, "Hacking techniques in wireless networks." [Online]. Available: http://www.cs.wright.edu/∼pmateti/InternetSecurity/Lectures/WirelessHacks/Mateti-WirelessHacks.htm

[19] "ARP request replay attack." [Online]. Available: http://www.aircrack-ng.org/doku.php?id=arp-request_reinjection

[20] Y.-M. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. T. King, "Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities." in *NDSS*, 2006.

[21] "Honeyclient project." [Online]. Available: http://www.honeyclient.org/trac

[22] 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.

[23] "Nmap network security scanner." [Online]. Available: http://insecure.org/nmap/

[24] "p0f: a versatile passive os fingerprinting tool." [Online]. Available: http://lcamtuf.coredump.cx/p0f.shtml

[25] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," in *USENIX-SS'06: Proceedings of the 15th conference on USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2006, pp. 12–12.

[26] A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless lan for the unlicensed band," *Bell Labs Technical Journal*, 1997.

[27] J. Bicket, "Bit-rate selection in wireless networks," Master's thesis, Massachusetts Institute of Technology, 2005.

[28] C. Corbett, R. Beyah, and J. Copeland, "Passive classification of wireless nics during rate switching," *EURASIP J. Wirel. Commun. Netw.*, To appear.

[29] N. Patwari and S. K. Kasera, "Robust location distinction using temporal link signatures," in *MobiCom '07*, 2007, pp. 111–122.

[30] "Powerful TCP port scanner, pinger, resolver." [Online]. Available: https://foundstore.com/resources/proddesc/superscan.htm

[31] "IP personality: a netfilter module to change characteristics of network traffic." [Online]. Available: http://ippersonality.sourceforge.net/

[32] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, J. Zahorjan, and E. Lazowska, "CRAWDAD data set uw/sigcomm2004 (v. 2006-10-17)," Downloaded from http://crawdad.cs.dartmouth.edu/uw/sigcomm2004, Oct. 2006.

[33] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 user fingerprinting," in *MobiCom '07*, 2007, pp. 99–110.

[34] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *SIGMETRICS '05*. New York, NY, USA: ACM, 2005, pp. 50–60.

[35] "Madwifi wlan device driver." [Online]. Available: http://madwifi.org/

[36] "Kismet: an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system." [Online]. Available: http://www.kismetwireless.net/

[37] "Aircrack-ng: an 802.11 WEP and WPA-PSK keys cracking program." [Online]. Available: http://aircrack-ng.org/doku.php

[38] "Advanced 802.11 attack: Black hat 2002." [Online]. Available: http://www.blackhat.com/presentations/bh-usa-02/baird-lynn/bh-us-02-lynn-802.11attack.ppt