

RAP: Protecting Commodity Wi-Fi Networks from Rogue Access Points

Liran Ma
Department of Computer
Science
The George Washington
University
Washington, DC 20052, USA
lrma@gwu.edu

Amin Y. Teymorian
Department of Computer
Science
The George Washington
University
Washington, DC 20052, USA
amin@gwu.edu

Xiuzhen Cheng^{*}
Department of Computer
Science
The George Washington
University
Washington, DC 20052, USA
cheng@gwu.edu

Min Song
Department of Electrical and
Computer Engineering
Old Dominion University
Norfolk, VA 23529, USA.
msong@odu.edu

ABSTRACT

We first give a comprehensive taxonomy of rouge access points (APs), which includes a new class of rouge APs never addressed in the literature before. Then, we propose an efficient rouge AP protection system termed as RAP for commodity Wi-Fi networks. In RAP, novel techniques are introduced to detect rouge APs and to improve network resilience. Our system has the following nice properties: i) it requires neither specialized hardware nor modification to existing standards; ii) the proposed mechanism can be integrated with an AP 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 can protect the network from adversaries capable of using customized equipment and violating the IEEE 802.11 standard.

1. INTRODUCTION

The increasing popularity of wireless networks has provided the network security field with many unprecedented challenges. For example, a new and significant security threat is the prevalence of rouge APs. Commodity Wi-Fi networks are particularly vulnerable to rouge APs because of factors such as open medium, insufficient software implementations, potential for hardware deficits, and improper configurations. According to a study by Gartner [7], rouge APs are present on about 20% of all enterprise networks.

A rouge AP is typically referred to as an unauthorized AP in the literature. This type of device can be easily deployed by end-users.

^{*}This research is supported by the National Science Foundation grant CCF-0627322.

When a rogue device 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 rouge APs as well. This paper also introduces a more insidious type of rouge AP, called the compromised AP, that has never been addressed before. A detailed taxonomy of Rouge APs is given in Section 3.

Advances in hardware and software have made AP discovery (e.g., finding unauthorized or improperly configured APs) an easy task for computer attackers. Commodity Wi-Fi network cards that have the capability to capture all 802.11 transmissions can be purchased for about US \$30 on eBay. Once in possession of a card, all that an attacker has to do is to find a program such as “iwconfig” in Linux that can enable this feature. Hence, the process of driving around and looking for vulnerable APs (known as “wardriving”) can be accomplished by people with limited security backgrounds. Moreover, 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 rouge AP. As shown in [17, 18, 21], the most common security protocol, Wired Equivalent Privacy (WEP), has been shown to be breakable even when correctly configured. Specifically, WEP fails to achieve any of the fundamental security goals of confidentiality, integrity, and availability.

Wi-Fi Protected Access (WPA) was created in response to the 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). 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 fails. Another deficiency of WPA is that it still relies on the RC4 encryption algorithm.

Due to these weaknesses in WEP and WPA, an AP can be easily compromised. Subsequently, the traditional way of protecting networks with encryption and firewalls is no longer sufficient. A novel system for protecting networks from rogue APs is needed to counter these security threats.

The contributions of this paper are twofold. First we provide a detailed classification of rogue APs, which includes a new class of AP never discussed in previous work. Secondly, we develop a novel rogue AP protection system called RAP that targets commodity Wi-Fi networks. RAP includes three major components: a packet collector, a rogue AP preemption engine, and a rogue AP detection engine. The proposed system can be connected to or implemented on APs as small plugins. It 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 violating the IEEE 802.11 standard.

The rest of this paper is organized as follows. Section 2 discusses related work. In Section 3, a comprehensive rogue AP taxonomy is presented. The proposed rogue AP protection system is elaborated in Section 4. Finally, our conclusion and future research directions appear in Section 5.

2. 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 [2–4]. AirDefense [2] is one such product. 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. Lastly, if the specialized monitoring sensors are not used, it is difficult to guarantee 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 [13]. Multiple APs and mobile clients perform RF monitoring to help detect the presence of rogue wireless devices like 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, RAP 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.* [14] 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 [13] and [14] assume that some specific characteristics of IEEE 802.11 standards cannot be violated by the adversaries. Conversely, RAP 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 [16] 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. RAP differs from this because it is completely automated and it provides comprehensive identification of rogue APs anywhere on a network.

Multiple network sniffers are used in [19] for detecting rogue APs and eavesdroppers. Each sniffer has three network cards, and the intrusion detection capabilities are stymied by MAC address spoofing. RAP 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.

Characteristics and network usage statistics of IEEE 802.11 WLAN in various settings are examined in [15] and [27]. Information from wireless link properties, TCP fingerprints, and SNMP scans is important in assisting with the prevention and detection of rogue APs.

Sometimes rogue AP detection functionalities are integrated into an intrusion detection system (IDS). A typical IDS scans network traffic and generates an alert when an intrusion has been detected. When used with intrusion prevention techniques such as realtime traffic flow analysis and automatic attack prevention, the security of a network is further enhanced.

Yeo *et al.* [29] 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, RAP is not limited to layer 2 traffic. It also supports automatic preemption and detection of various network attacks.

An AP-based system called DOMINO is proposed in [26] 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 [23], 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.

RAP differs from previous work in that it provides robust and comprehensive protection against rogue APs for commodity Wi-Fi networks. It also defends against a new class of rogue APs that has never been discussed in the literature before. Further, it can detect rogue APs that have the ability to violate the IEEE 802.11 standard. RAP improves the resilience of Wi-Fi networks through an elegant coupling of rogue AP preemption and detection. Moreover, the mature techniques and freely available software that RAP employs make it an efficient and cost-effective solution. Lastly, modi-

fications to the underlying wireless standard are not necessary with RAP.

3. ROGUE AP TAXONOMY

Rogue APs are typically placed into one of three categories: improperly configured, phishing, and unauthorized. In this section, we explain each of the above rogue AP classes, and introduce a more insidious type of rogue AP called the compromised AP. Our classification system appears below.

3.1 Improperly Configured AP

Even without malicious intent, a legitimate AP can suddenly turn into a rogue device because of a minor configuration mistake. There are several scenarios where an AP can be improperly configured. A network administrator with insufficient security knowledge (e.g., an inability to choose appropriate authentication and encryption settings) could fail to set up the AP properly. It is also possible that an AP's driver is faulty or that the device itself is physically defective. An even worse case occurs when a properly configured AP becomes vulnerable after a software update. For example, a driver or firmware update to an AP with WEP enabled could cause it to restart without WEP protection.

Additionally, it is possible that a computer has both a wired interface and a wireless interface. A network tap can be connected to the wired card, and the wireless card can be placed in ad hoc mode. An ad hoc connection usually lacks the necessary security measures such as 802.1X user authentication. As a result, the computer can provide AP (or gateway) functionality to an attacker that creates a wireless ad hoc connection to it. A properly specified and implemented security policy can avoid this vulnerability. Determining proper security policy is beyond the scope of the paper.

3.2 Unauthorized AP

Installing an AP on a secure network without authorization from the network administrator also creates a rogue AP. Even though the AP is not managed by the network administrator, it can still become accepted as part of the official network. This is because the AP transmits and receives network traffic as does a legitimate AP. Driven by the convenience of network access, this class of rogue APs routinely exists in large organizations with many employees. Anyone with physical access to the premises can ignorantly or maliciously connect a cheap wireless AP to network. Subsequently, novice users have the ability to create large vulnerabilities to enterprise security and expose an otherwise-secure corporate intranet to unauthorized parties. Casual visitors or malicious hackers can connect to the unauthorized AP to steal bandwidth, retrieve confidential data, attack company assets, or use the network to attack others.

Another possible origin of rogue APs is neighborhood WLANs. According to the 802.11 standard, clients prefer to connect to an available AP with high signal strength. For example, Windows XP can automatically connect the best connection available within communication range. Due to this behavior, authorized clients of one organization may inadvertently connect to APs from a neighboring organization. Though a neighboring AP has not intentionally lured a client into connecting with it, these associations can still expose sensitive data.

3.3 Phishing AP

An attacker can set up an AP outside the wireless network of a facility. It attempts to fraudulently acquire critical credentials, such as usernames, and passwords, by masquerading as a trustworthy AP. In addition, it can be configured to replay beacons that it overhears from legitimate APs, thus fooling some clients within the facility to connect to it. This can allow an attacker to conduct a man-in-the-middle (MITM) attack on a Wi-Fi network that does not enforce mutual authentication (i.e., client-to-server and server-to-client authentication). It is quite common that WEP-enabled networks do not employ mutual authentication.

There are also easy-to-use and freely available tools such as FakeAP [6] that allow an individual device to masquerade as multiple APs. Although intended to obfuscate a network's presence to wardrivers, the software can also be used to confuse legitimate users of networks with similar SSIDs. An attacker may use this tactic in conjunction with an AP outside the target network to facilitate the launch of a MITM attack.

3.4 Compromised AP

Even if an AP is properly configured with security features like WEP or WPA-PSK enabled, an attacker can still crack the key being used. With the advances in hacking software, a person with very little 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 [5] 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 [1].

Once the secret key is discovered by an adversary, all of the APs using the same credentials in a Wi-Fi network become rogue APs. This reflects a significant payoff for the adversary because a greater number of physical locations for launching attacks become available. Further, a compromised AP allows an attacker to easily masquerade herself as a legitimate user and gain access to potentially sensitive data.

A seemingly easy way to avoid AP compromise is to upgrade the security protocols used in WPA2. However, WPA2 is not compatible with the legacy hardware that is widely used in commodity Wi-Fi networks. Furthermore, a large-scale equipment upgrade would incur prohibitively high cost.

3.5 Remarks

As mentioned in Section 2, it is possible to detect rogue APs that fall into the first three classes by performing a walking audit around the facility of interest. Alternatively, probes that constantly monitor the wireless network looking for changes or a server that monitors both wired and wireless sides of a network can be used. However, the fourth class of rogue APs cannot be dealt with as easily.

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 Class 4 rogue AP can significantly diminish the overall security of the network.

RAP is our solution for preempting attacks that can create rogue APs, and for detecting the presence of such devices when they exist. By taking advantage of APs and wireless range extenders instrumented with free software, RAP also provides cost-effective

protection of commodity Wi-Fi networks. A summary of the types of rogue APs that RAP defends against is shown in Table 1.

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 1: Rogue AP Taxonomy and Scenarios.

4. THE RAP SYSTEM

RAP is designed to monitor network activities, forestall events that could lead to the generation of rogue APs, block unauthorized network access through rogue APs, and eliminate existing rogue APs. The three main components that constitute the RAP architecture are: a packet collector, a rogue AP preemption engine, and a rogue AP detection engine. An illustration of the overall architecture of RAP can be seen in Fig 1.

The packet 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.

These components can be implemented on an AP or on separate devices that connect to the AP in a plugin 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 three components are implemented on it. The details of each component are described in the following subsections.

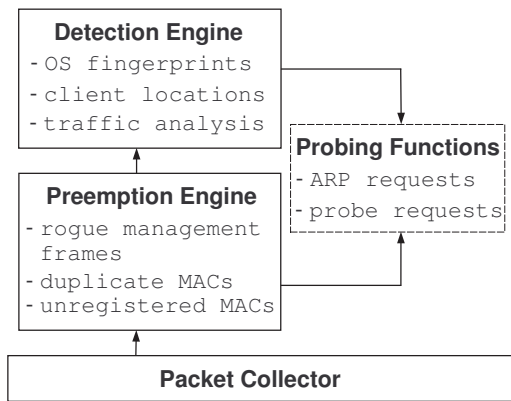


Figure 1: The software architecture of RAP.

4.1 Assumptions

The type of Wi-Fi network that we consider uses WEP or WPA in conjunction with MAC address filtering. This combination of features is prevalent in commodity Wi-Fi networks. 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.

Some research argues that monitoring a network from devices such as APs cannot provide comprehensive coverage [14]. 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. Based on the specifications of an off-the-shelf extender (e.g., the Belkin F5D7132), these types of devices can scan all working channels of 802.11b and 802.11g. Additionally, they offer a working range up to 457.2 meters.

We assume that a floor plan of the building containing the to-be-protected network is available to RAP. In particular, the exact location of authorized APs and range extenders should be known by RAP. 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 RAP.

4.2 Packet Collector

A packet 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 a packet 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.

4.2.1 Information Collection

The packet 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 packet collector’s duties is to capture all network traffic. In order to measure the storage overhead, we conducted a test using our department (802.11 b/g compatible) Wi-Fi network. The test consisted of a single 802.11b client using ftp to download a large file from a local server. We carried out the test in the middle of the night so that there would be no contention from other wireless clients. The average transfer rate recorded over many trials was approximately 2.2 Mbps. Therefore, by recycling the collected data every one hour, the storage overhead can be limited to about 1 Gigabyte. This is a reasonable overhead for even low-end computing equipment.

Additionally, the packet 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. Lastly, the measured RSSI values from both APs and extenders are provided to the detection engine.

Note that it is important to hide the packet collector from adversaries. Otherwise, a prudent attacker may change tactics to elude capture. To prevent inadvertently revealing its presence, RAP performs all sniffing in a passive manner. However, achieving real passive is not trivial. 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 eavesdropping device.

In actuality, there are a number of ways that a “passive listener” can actually be 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: 1) the Address Resolution Protocol (ARP), a protocol that is primarily used for translating IPv4 addresses to Ethernet MAC addresses; 2) an extension to ARP called Inverse Address Resolution Protocol (InARP) that performs the inverse of ARP in Frame Relay networks; 3) the Bootstrap Protocol (BOOTP), which allows a client to obtain an IP address automatically during the bootstrap process.¹

In the following, we propose techniques that achieve real “passive listening.” We detail these techniques below.

4.2.2 Methods to Achieve Passive Listening

The first technique we describe 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.

Reconfigure Network Card: One way to prevent responding to any message that uses IP related information² 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 can 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.

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 device, such as a packet collector, needs to have the above protocols disabled as well.

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

²This includes both direct and indirect use of IP information, such as a ping message and an ARP request, respectively.

4.3 Rogue AP 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 with RAP.

The rogue AP preemption engine of RAP 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 a legitimate AP from being compromised.

4.3.1 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 “passively listening” to the network traffic and replies to one of the trap ARP requests, her presence will be revealed.

The interval selected for broadcasting these 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 systems.

4.3.2 Attack Preemption

After obtaining data from the packet collector, the rogue AP preemption engine will perform the checks outlined below. By preempting attacks that could reveal the secret network key, we prevent the creation of Class 4 rogue APs.

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., [20]). 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. Additionally, such information will be shared with the rogue AP detection engine described in Section 4.4.

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 [25]. 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 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.

Nevertheless, there are some cases where an attacker might go unnoticed by our preemption system. For example, the attacker might choose to employ the passive listening techniques described in Section 4.2. 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 go unnoticed by our integrity check module, RAP includes the rogue AP detection capabilities described in the next subsection.

4.4 Rogue AP 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. Secondly, a sophisticated adversary may be able to evade the preemption techniques for Class 4 rogue APs.

The rogue AP detection engine is responsible for discovering rogue APs regardless of what class they belong to. For Classes 1 – 3, the AP probing technique described in Section 4.4.1 is used to lure rogue APs into revealing their presence. 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 detail the steps used by RAP’s detection engine below.

4.4.1 AP Probing

An AP advertises its presence several times per second by broadcasting special frames that carry its SSID called beacons. 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 functionality³ to discover rogue APs by sending out probe requests. It is capable of detecting the first three classes of rogue APs.

There is a common misconception that disabling the “Broadcast SSID” feature hides the SSID. In reality, disabling this feature only makes the AP transmit a null (zero-length) SSID in beacon frames and probe responses instead of its 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.

Therefore, a particular AP can be discovered from its probe responses. The next step is to determine whether or not it is a rogue

³Examples of active honeypot systems include Strider HoneyMonkeys [28] and the Honeyclient Project [8].

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 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 RAP. 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, RAP can have the switch port that is associated with the rogue AP’s MAC address 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.

RAP 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. As a result, there is no MAC address information available to draw a conclusion. Nevertheless, a disassociation message can be sent to a client of the suspect AP based on the stream information collected by the packet collector. When the client sends out a reassociation request, the MAC address and SSID of the 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 [13] and [14] 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. RAP 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.

4.4.2 Compromised AP Detection

A compromised AP is detected by identifying an unauthorized client who is connecting to it. The client is detected by employing a combination of MAC address information and OS fingerprinting techniques. Recall that MAC address filtering is used by the networks under the protection of RAP.

The first 3 bytes of a MAC address, the Organizationally Unique Identifier (OUI), allow us to determine the assignee of a particular card company [9]. 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 potentially 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 [11] and p0f [12], respectively. With this 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 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.

It is also possible for a sophisticated attacker to 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. A freely available tool that performs such functions is IP Personality [10]. Still, there is no guarantee that an attacker knows the OS that typically runs on the machine with (the network adapter with) the MAC address she is spoofing. Then, a guess should be made as to what OS a client is using based on OS market share information. If the guess is correct, the attacker cannot be identified. In this case, intrusion detection based on techniques from machine learning and data mining [24] could be performed. For example, a profile could be created for each client that indicates their Internet usage characteristics.

5. CONCLUSION

In this paper, we provided a comprehensive taxonomy of rogue APs. Our classification includes improperly configured APs, phishing APs, unauthorized APs, and a new class of rogue AP termed as compromised APs. We then develop a novel system for protecting commodity Wi-Fi networks from rogue APs called RAP. An attractive feature of RAP is that it requires neither specialized hardware nor modification to existing security standards. Further, the proposed mechanism 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, RAP 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. RAP is the first system that can successfully protect the network under that assumption.

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

6. REFERENCES

- [1] Aircrack-ng: an 802.11 wep and wpa-psk keys cracking program.
- [2] AirDefense Enterprise: a wireless intrusion prevention system.
- [3] AirMagnet: Enterprise wlan management.
- [4] AirWave: Wireless network management.
- [5] BackTrack live cd by remote exploit.
- [6] Fake AP: Counterfeit AP generation tool.
- [7] Gartner advises on security.
- [8] Honeyclient project.
- [9] IEEE RA frequently asked questions.
- [10] IP Personality: a netfilter module to change characteristics of network traffic.
- [11] Nmap: Network mapper.
- [12] p0f: a versatile passive os fingerprinting tool.
- [13] A. Adya, P. Bahl, R. Chandra, and L. Qiu. Architecture and techniques for diagnosing faults in ieee 802.11 infrastructure networks. In *MobiCom '04*, pages 30–44. ACM Press, 2004.
- [14] 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*, pages 1–14. ACM Press, 2006.
- [15] 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*, pages 195–205. ACM Press, 2002.
- [16] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland. Rogue access point detection using temporal traffic characteristics. In *GLOBECOM*, 2004.
- [17] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: the insecurity of 802.11. In *MobiCom '01*, pages 180–189. ACM Press, 2001.
- [18] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker. Security flaws in 802.11 data link protocols. *Commun. ACM*, 46(5):35–39, 2003.
- [19] M. K. Chirumamilla and B. Ramamurthy. Agent based intrusion detection and response system for wireless lans. In *ICC '03*, pages 492–496, 2003.
- [20] 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).
- [21] 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*, pages 1–24, London, UK, 2001. Springer-Verlag.
- [22] C. He and J. C. Mitchell. Security analysis and improvements for ieee 802.11i. In *NDSS*, 2005.
- [23] P. Kyasanur. Selfish mac layer misbehavior in wireless networks. *IEEE Transactions on Mobile Computing*, 4(5):502–516, 2005. Senior Member-Nitin H. Vaidya.
- [24] M. A. Maloof. *Machine Learning and Data Mining for Computer Security: Methods and Applications (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [25] P. Mateti. Hacking techniques in wireless networks.
- [26] M. Raya, J.-P. Hubaux, and I. Aad. Domino: a system to detect greedy behavior in ieee 802.11 hotspots. In *MobiSys '04*, pages 84–97. ACM Press, 2004.
- [27] D. Schwab and R. Bunt. Characterising the use of a campus wireless network. In *INFOCOM*, 2004.
- [28] Y.-M. Wang, D. Beck, X. Jiang, R. Rouseff, C. Verbowski, S. Chen, and S. T. King. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In *NDSS*, 2006.
- [29] J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless lan monitoring and its applications. In *WiSe '04*, pages 70–79. ACM Press, 2004.