This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

Catch Me If You Can: Rogue Access Point Detection Using Intentional Channel Interference

Rhongho Jang, Jeonil Kang, Aziz Mohaisen, *Senior Member, IEEE,* DaeHun Nyang, *Member, IEEE.*

Abstract—In this paper, we introduce a powerful hardware-based rogue access point (PrAP), which can relay back and forth traffic between a legitimate AP and a wireless station, and act as a man-in-the-middle attacker. Our PrAP is built of two dedicated wireless routers interconnected physically, and can relay traffic rapidly between a station and a legitimate AP. Through experiments, we demonstrate that the state-of-the-art time-based rogue AP (rAP) detectors cannot detect our PrAP, although perhaps effective against software-based rAP. In demonstrating that, we unveil new insight into fundamentals of time-based detectors for software-based rAPs and their operation: such techniques are only capable of detecting rAPs due to the speed of wireless AP bridging. To address the threat of such PrAPs, we propose a new tool for network administrators, a PrAP-Hunter based on intentional channel interference. Our PrAP-Hunter is highly accurate, even under heavy traffic scenarios. Using a high-performance (desktop) and low-performance (mobile phone) experimental setups of our PrAP-Hunter in various deployment scenarios, we demonstrate close to 100% of detection rate, compared to 60% detection rate by the state-of-the-art. We show that our PrAP-Hunter is fast (takes 5-10 seconds), does not require any prior knowledge, and can be deployed in the wild by real world experiments at 10 coffee shops.

Index Terms—Intrusion detection, Wireless LAN, Rogue AP, channel interference, IEEE 802.11n.

1 Introduction

THE Wireless Local Area Network (WLAN) as a technology is popular in part for supporting mobility, a feature that makes WLAN deeply integrated in many essential applications to facilitate an easy Internet access at public spaces such as restaurants, cafes, and public libraries. Due to the mobility features in WLAN, their easy setups, and the lax use policies in many deployment scenarios, various security threats have emerged. For example, while WLAN allows users to easily set a new wireless Access Point (AP) up using off-the-shelf WLAN hardware and their electronic gadgets (e.g., laptop computer or smartphone), that same feature also allows an adversary to set up a rogue AP (rAP), for potentially attacking benign users. Even worse, many WLAN users are unaware of the security dangers associated with wireless access, especially when connecting to APs in public places.

With many public spaces, including shopping malls, restaurants, and public transit systems, providing WLAN services and power outlets for customers, an adversary equipped with a laptop and an additional network interface can easily create a *persistent* rAP to eavesdrop on, intercept, or even modify communications between users and the Internet. An adversary capable of creating such rAP can use it to launch a large array of attacks on innocent users connecting to it. For example, the attacker can eavesdrop on the exchange of sensitive information such as identity credentials, password, and bank account by observing relayed packets as shown by Brenza *et al.* [2]. The attacker can also mount an active attack by rewriting DNS queries and response to lead users to phishing websites. The attacker can even infect the user's device with a malicious software (malware) by reflecting malicious contents in response to the user's browsing requests.

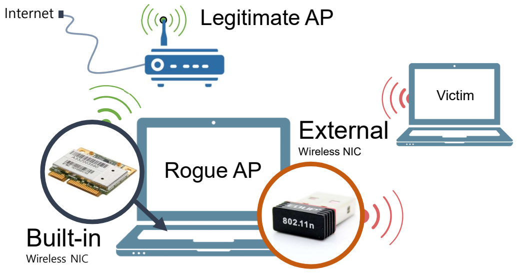


Figure 1. General rogue Access Point (rAP) setup using a laptop with a built-in and an external wireless interfaces.

Indeed, various recent research studies have pointed out and experimentally demonstrated this security issue as a threat [3], [4], [5], [6], [7]. Also, several research results showed that threats caused by the rogue AP were quite practical even using TLS and SSL [8], [9], [10]. Moreover, various recent media reports (from 2013 to 2016) have detailed real attack incidents using rAPs, and stressed various security, privacy, and public safety implications [11], [12], [13], [14], [15], [16]. For example, in late 2013, researchers from Trend Micro™ conducted a rAP experiment in the city of London, where they found that: 1) users deliberately connected to such AP for various online activities, and 2) users liberally exposed their information to the attacker, including login credentials, transactions, and other sensitive information [16].

As in the literature, a rAP is typically created using two wireless interfaces, where the first interface is built-in while the second one is external as shown in Figure 1 as shown in [17]. The built-in interface (e.g., inside a laptop) is operated in the station mode and used for connecting to a legitimate AP, while the external interface is operated in the service mode and used by users as trusted AP. The two interfaces forward packets to each other and relay Internet services on behalf of the user (victim). In

D.H. Nyang is the corresponding author. R. Jang, J. Kang and D.H. Nyang are with the Department of Computer Engineering, Inha University, Incheon 22212, Korea. R. Jang (second affiliation) and A. Mohaisen are with the Department of Computer Science at the University of Central Florida, FL, USA. (e-mail: r.h.jang@knights.ucf.edu, dreamx@isrl.kr, mohaisen@ucf.edu, nyang@inha.ac.kr). A preliminary version of this work has appeared in IEEE ICDCS 2017 [1].

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

this scenario, a rAP is created using the same Service Set Identifier (SSID) of a cloned victim AP to perform a man-in-the-middle attack, also known as the evil-twin. The evil-twin is not necessarily the clone of the legitimate AP providing Internet connectivity, and could be any AP in the vicinity for the attacker to clone—in this paper, and for consistency and clarity, we use "*legitimate AP*" to refer to an AP that provides Internet connectivity, and "*trusted AP*" to refer to a victim cloned by an attacker.

The recent research efforts and media reports only highlight its prevalence today, and the rAP attack has been known for many years [18]. As such, there have been various attempts to defend against it resulting in various detectors, including two notable classes of rAP detection techniques: the snooping-based [19], [20], [21], [22] and time-based approaches [3], [5], [23], [24], [25]. In the snooping-based approaches, Media Access Control (MAC) or SSID addresses of rAPs are collected in a blacklist and used later to detect them. However, this approach is fundamentally limited, since MAC and SSID addresses can be easily spoofed. On the other hand, the time-based approaches, as in [3] and [5], use the timing side-channel to detect rAPs. They assume that when packets are sent through a rAP, a relaying delay is observed, because the rAP has to use an additional wireless path for its operation. Indeed, in validating such assumption, the prior work used software-based rAPs and demonstrated a significant delay when using an AP. It has been widely accepted without validation that the observed delay is the result of the additional wireless path.

In this paper, we illustrate a limitation of the time-based techniques by showing that the delay used for inferring whether a rAP exists between a user and a legitimate AP is not due to an additional wireless path, but the result of a computational delay caused by the software bridging. We demonstrate that an adversary can manipulate this delay feature and evade detection by adopting a high-performance hardware-based *layer-2* wireless bridge with minimal bridging delay. We devise a new detection technique and demonstrate its effectiveness in detecting the proposed hardware-based rAP under the assumption that a rogue AP should use two different channels (one for relaying a legitimate AP and the other for serving stations), The assumption is verified in §3. Our detector uses two wireless interfaces: one sends a steady flow of traffic via the target AP to a remote server, while the other intentionally interferes with other channels one by one. When the target AP is legitimate, traffic obstruction caused from the interference is not observed. If our detector connects to a rogue AP using two different channels, we can observe traffic obstruction to rAP even when the interfering device is working at the other channels.

Contributions. The contributions of this work are multifold. (1) We developed a powerful rogue AP (PrAP) that defeats the existing time-based detectors and show their fundamental shortcomings. Different from the existing rAP designs using a laptop and a wireless adapter, PrAP consists of two physically interconnected off-the-shelf WLAN routers. We implemented a time-based detector [3], tested it, and showed how it fails to detect PrAP. (2) We designed PrAP-Hunter, a new detector based on a new detection assumption, namely the channel interference. PrAP-Hunter is a tool for network administrators to determine whether a given and currently connected AP is a rAP or not with high accuracy. Through extensive experiments, we show that the wireless channel communication can be interfered by intentional channel interference signals, and we quantify the throughput degradation according to the amount of channel interference. We also show that the intentional interference can be used not only to perform attacks but also to counter-intuitively and effectively defend attack from a rogue AP (e.g., rogue AP detection). (3) To the best of our knowledge, our work is the first scheme that considers channel configuration issues of the wireless bridged rogue AP. Our system detects rogue APs that are set up in a wide range of channel settings unlike previous works that can only detect attackers serving in a channel far from that of the legitimate AP. (4) We implemented PrAP-Hunter in two different hardware setups: a desktop computer and a mobile phone. We performed an extensive evaluation under various traffic scenarios using the desktop detector, and under different locations (positions) using the mobile detector. We found that PrAP-Hunter achieved close to 100% detection rate with the desktop setup, regardless of the traffic scenario. With the mobile setup, the detection rate was 100% when PrAP-Hunter was located close to the PrAP. Performance-wise, PrAP-Hunter provides a significant improvement over the state-of-the-art: Han *et al.*'s achieved 60% detection rate under heavy traffic scenarios with a software rAP, whereas PrAP-Hunter's rate is significantly higher even under a worse scenario [3]. We supplement our work with a field study for detection at 10 coffee shops. Using the mobile version of PrAP-Hunter, we executed a successful detection in all cases.

PrAP-Hunter has several advantages. (i) It can detect a hardware-based rAP that cannot be detected using time-based rAP detectors. (ii) It works without requiring any prior knowledge of information such as the SSID, MAC address, Received Signal Strength Indicator (RSSI), and clock skew on the examined network. (iii) It provides significantly higher detection rates even under a heavy traffic scenario. (iv) It is fast, and the detection of a rAP is completed within 10 seconds. (v) It is cheap; implemented on a smartphone with an additional off-the-shelf WLAN card.

Organization. The organization of the rest of this paper is as follows. The related work is described in §2. The threat model is outlined in §3. The detection strategy is outlined in §4. Our detector for rogue APs is described in §5. Our experimental setup is described in §6. In §7, we present our experimental results and performance evaluation. Real-world deployment and testing are addressed in §8, while concluding remarks are drawn in §10.

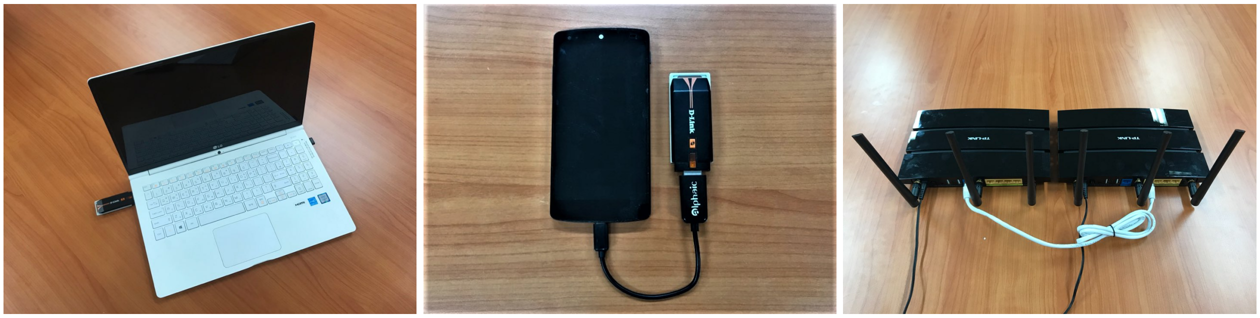
2 Related Work

Rogue AP (rAP) detection methods are mainly classified into two categories: snooping-based [19], [20], [21], [22], [26], [27], [28], [29], [30] and time-based detection [3], [5], [23], [24], [25]. The snooping-based schemes use sensors to collect features of APs, e.g., SSID, MAC address, channel, RSSI, and clock skew. The collected features are then compared with previously known features of rogue (or legitimate) APs to determine the legitimacy of a given AP. The second category of schemes depends on the characteristics of inter-packets, the round trip time or traffic to detect rAPs. Generally, those techniques do not require any prior knowledge about the wireless devices, but sometimes they need to configure site-specific parameters for better detection rate. These schemes can actively detect a rAP by collecting the required information in real time.

Snooping-based approaches. In the snooping-based approaches, a prior knowledge is used to detect the presence of a rAP. In [19], [21], [22], the MAC address of an AP is compared against addresses of known APs for detection. An unknown MAC address indicates that an AP is rogue. Also, other nonforgeable factors like RSSI values [27], [29], clock skew [28], [30], or radio frequency

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing



(a) Laptop/USB

(b) Phone/USB

(c) WAN/WLAN (PrAP)

Figure 2. Three ways ways of setting up a rogue AP (rAP), using a universal serial bus (USB)-based wireless interface (along with the built-in wireless interface) as in (a) with a laptop and (b) with a phone, and using a physical layer connector of two routers (PrAP) as in (c).

variations [26] are used to fingerprint rAPs. While easy to use as detection features, it is well known that features such as identifiers, including the SSID and MAC address, can easily be spoofed. Further, those approaches are costly in requiring equipment setup for collecting such a prior knowledge (*i.e.,* authorized list), or collecting data (*e.g.,* RSSI, RF wave) from traffic sensors.

Time-based approaches. Beyah *et al.* suggested a method that utilizes temporal characteristics, such as inter-packet arrival time [24]. Wei *et al.* [31], [32] proposed two similar detection schemes by examining the arrival time of consecutive ACK pairs in TCP traffic. Watkins *et al.* and Qu *et al.* [33] used the round trip time of TCP traffic, Venkataraman *et al.* [34] based their approach on DCF pattern in the wired traffic, while Mano *et al.* [35] based their approach on physical properties of half duplex channels for detection. All the above works are for detection of wired rogue APs, but for wireless rogue APs, there are two known works: one is by Yang *et al.* [5], and the other by Han *et al.* [3]. Yang *et al.* proposed an "evil twin" detector using a discriminative feature of inter-packet arrival time of a rAP [5]. Han *et al.* developed a time-based detection technique that uses RTTs through additional wireless line [3]. These two techniques use packet delay of traffic caused by the rAP as a feature for detection. While they have a lower cost than snooping-based approaches, since they do not require any setup of any additional sensors, these schemes are sensitive to network conditions, with network instability causing spikes in the false alarm rates.

Other approaches. Lanze *et al.* [7] fingerprinted software rAPs using attack tools properties (e.g., MadWifi [36], aircrack-ng [37], Karma [38]). Kindberg *et al.* [39] and Roth *et al.* [40] proposed two authentication-based approaches that additionally require the user interaction. In [39], users required to check key management results shown on an additional display belonging to the legitimate AP. Similarly, in [40], the established key was used to encode a short string as a sequence of colors, and rendered in both the user device and the legitimate AP. While efficient, these works do not support multiple users authenticate at the same time. Bauer *et al.* [41] suggested to mitigate evil twin attacks by using the contextual information which is defined by the information of nearby APs. This work was based on Trust on First Use (TOFU), which has potential vulnerabilities when associating with an AP for the first time. Later, Gonzales *et al.* [42] improved Bauer *et al.*'s work by combining the RSSI with the context. Moreover, they adapted Pang *et al.*'s "wifi-report" system [43] to reduce the risk of TOFU. Gonzales *et al.* also suggested an SSH-like authentication method to secure data delivery. However, modifications of EAP protocol were required. Our threat model is more realistic by not requiring TOFU or infrastructure modification. To this end, the literature focused on software-based rAPs, and proposed detections for them. In this paper, we introduce a powerful hardware-based rogue access point (PrAP) that can evade time-based rAP detectors, by avoiding the timing channel, and evade the snooping-based methods by spoofing their detection information (e.g., MAC and SSID), and by turning off the broadcast of beacon frames. To address this PrAP, we propose a channel interference-based PrAP-Hunter. Channel interference is a kind of jamming, which has been treated to be defeated as in [44], but we use the jamming in a positive way to detection.

3 Threat Model

A network administrator needs to check whether an AP in the enterprise network is a trusted AP or a rogue AP (rAP). Regular check-up of rAPs are desirable because users carry out confidential communication over an AP that they believe trustful. In this paper, a rAP is defined as an AP that relays WLAN traffic between *a legitimate AP* providing Internet connectivity and a station, and may act as a man-in-the-middle trusted AP of which device information is cloned from *a trusted AP*. To this end, we assume that a rAP has two wireless interfaces, one connected to the legitimate AP in station mode and another disguised as the trusted AP in service mode. When a user connects to the rAP, two interfaces will forward traffic back and forth. This relaying attack has been reported in [17].

Software-based rAP. In the literature, rAPs are defined using a laptop and an additional WLAN USB adapter, as shown in Figure 2(a) [3], [4], [5], [7]. This type of rAP can easily be set up by adding rules to the `iptable` or by setting up Internet sharing functionality of Microsoft Windows or Mac OS. As shown in Figure 2(b), with the development of smart devices, we can setup a rAP by utilizing a mobile phone and an additional WLAN USB adapter, since some customized ROMs support WLAN connection with on-to-go (OTG) cable. Configuring a rAP with a laptop or a mobile phone can give an adversary portability features. However, such rAPs relay packets between two wireless interfaces in a software-based approach. Therefore, the performance of such APs depends on the computational power of the software bridging. We proved in §7 that the time-based approach proposed by Han *et al.*'s [3] can detect software-based rAPs only, but not the powerful hardware-based rAP (PrAP) having little bridging delay. Also, Lanze *et al.* [7] outlined an effective method to fingerprint such rAPs.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

Hardware-based PrAP. Figure 2(c) shows a setup of a PrAP costing under $100, and achieving high performance in relaying packets between two wireless interfaces in a hardware-based approach. The PrAP is characterized by a low delay, and is difficult to detect using time-based rAP detection methods. Moreover, the plentiful capacity (*i.e.,* 1 Gbps) of mirroring port of PrAP helps capturing raw packets and injecting manipulated packets without packet loss. We note that the mirroring port does not delay the packet relaying pipeline (See details in Section 6.1).

3.1 Assumptions

An attacker in our threat model is assumed to wirelessly connect to a legitimate AP (*i.e.,* Internet provider) and to clone all information of one of the trusted APs (*i.e.,* cloned target) except the channel information. The Internet provider can be one of the APs around the attacker, including the cloned target APs in the enterprise network or any other APs providing Internet connection. We consider various cloning scenarios to explain our assumption of the channel used by adversaries.

As a way of replacing the two interfaces with one interface, one may assume as well the ability of the adversary to plug a rAP directly in the backbone, thus eliminate the need for using a second channel, which is utilized in our approach for detection. First, we point out the large body of the literature [23], [24], [25], [31], [32], [33], [34], [35] that address the problem directly with wired rAPs. Moreover, we also point out that the assumption made is very strong and often impractical. A plugged rAP in most enterprises would be visible, and can be detected through physical security measures. Even where that is possible (*e.g.,* an employee unplugging his PC with a dedicated IP, and replacing it with a rogue access point), plausible scenarios leading to this kind of attack are out of the scope of our threat model: they would require an insider attacker, which is way beyond the attack capabilities we assume in our work.

3.1.1 Basic assumption

Our basic assumption is that adversaries clone the SSID, MAC address and password of a target AP. There are several reasons for assuming and justifying how adversaries can clone the password of the target APs. First, this assumption is necessary for the operation of the rogue AP. For example, if the rogue AP is to use a different password than the one known to users using a public (legitimate) AP, connections by victims will be automatically rejected. While one can cope with this issue (*e.g.,* making the AP public or programmatically modifying the AP to accept any password), not needing to copy the password, such a mitigation would require modifying the AP. Moreover, when getting rid of the password altogether and making the AP public, recent operating systems (*e.g.,* Microsoft Windows, Apple's iOS and Google's Android) alert users about unsafe wireless connections when accessing public APs without authenticating. Second, the justification of being able to clone the password is quite straightforward in today's wireless access points usage. For example, in public spaces and facilities, such as restaurant, hospital, shops, public transportation, lounges, *etc.*), the WiFi password is often posted on the wall of the facility. As such, it is easy for an adversary to obtain the password of the legitimate AP. Finally, as mentioned in reasoning about the assumption, a determined adversary can build his own RADIUS server that approves all accesses without verifying the credentials, which has the equivalent effect of password cloning.

3.1.2 Cloning channel of Internet provider APs

An adversary should avoid cloning the channel of an Internet provider AP because of the co-channel interference problem. In our work, the rogue AP consists of two wireless interfaces. According to Villegas *et al.* [45], a channel sharing by two wireless interfaces degrades a 6 Mbps traffics by around 50% (*i.e.,* 3 Mbps), even when they are placed in different rooms (*i.e.,* when having a large spatial distance), and under light network conditions. Moreover, Zubow *et al.* [46] proved that the distance between the wireless interfaces is also an important factor affecting the interference. Per to their results, a smaller spatial distance (*e.g.,* less than 1 meter) between two wireless interfaces leads to a much stronger channel interference. Because the spatial distance for a rogue AP is usually limited (*e.g.,* they two interfaces have to be contained in proximity), the rogue AP must increase the channel interval between wireless interfaces for avoiding the self-interference.

3.1.3 Cloning channel of target APs

To observe what happens when a rogue AP clones all information of the target AP (*i.e.,* SSID, MAC address, channel, the security protocol (WPA, WEP), encryption (AES, TKIP), and the password), we conducted an experiment, where two smartphones running different operating systems (*i.e.,* Android and iOS) established wireless connections with those identical access points. We found that only one SSID could be probed by both smartphones, which means the PrAP could successfully disguise itself by cloning the information of the legitimate one. However, both smartphones eventually failed to establish a connection with either of the APs. By examining the wireless management frames (Layer 2), we found that the client failed in the four-way handshake step of establishing a key with the access point. The connection failure happens because all key exchange frames from the client are heard by both the rogue and the legitimate APs. In establishing a session key, we note that both the rogue and the legitimate APs reply back to the smartphone frames with different random keys (*e.g.,* K1 and K2). Then, the client replies back with the first frames (*e.g.,* K1) it received, leading the AP that sent the frame with K2 and received the wrong reply for K1 to send back a connection failure message to the station. In conclusion, the station cannot make the connection to the rogue AP on the same channel.

3.1.4 Channel use of PrAP (Stronger Adversary)

It is also a common practice when providing wireless services in a large area (*e.g.,* enterprise, campus, shop, *etc.*) for network administrators to set up several APs with the same SSID operating on different channels. Having multiple APs and running them on different channels (*e.g.,* 1, 6, 11) is important to cover a whole area with a strong RSS, to address separate workloads while avoiding channel interference. Moreover, multiple APs within close proximity (*e.g.,* three legitimate APs serving on channels 1, 6 and 11, and using the same SSID but different MAC addresses) can be forced to use the same channel and even the same MAC. However, as shown earlier, any connection cannot be established in this scenario as the four-way handshake procedure in the session key establishment would fail, which also would be the case for our rAP if operated on the same channel as one of the legitimate APs. As such, and in order to address this issue in a setting with multiple legitimate APs, the adversary would clone the information of the legitimate AP on channel 1, and serve on channel 6 or 11 by relaying through channel 1. As a result, rAP would still provide internet service without self-interference.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

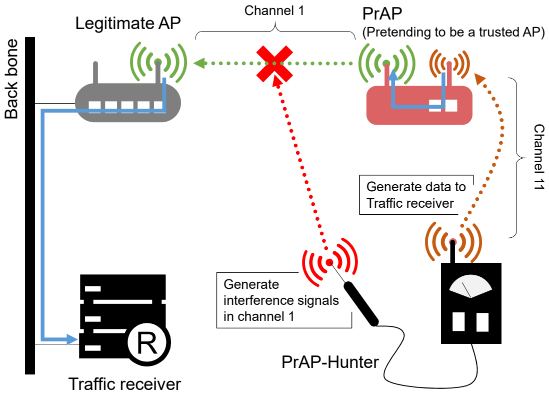

Legitimate AP
Channel 1
PrAP
(Pretending to be a trusted AP)
Back bone
Channel 11
Generate data to Traffic receiver
Generate interference signals in channel 1
PrAP-Hunter
R
Traffic receiver

Figure 3. A legitimate AP on channel 1 and a PrAP repeating signals of the legitimate AP on channel 11. The PrAP-Hunter generates traffic to the traffic receiver through the PrAP. The interference device interferes with channel 1. An AP is said to be rogue if we observe obstruction of traffic on channel 11 via the PrAP-Hunter.

Table 1
Bandwidth overlap against channel gap

Channel gap	Overlapping bandwidth
0 (ch1 vs. ch1)	20 MHz
1 (ch1 vs. ch2)	17 MHz
2 (ch1 vs. ch3)	12 MHz
3 (ch1 vs. ch4)	7 MHz
4 (ch1 vs. ch5)	2 MHz
5 (ch1 vs. ch6)	0 MHz

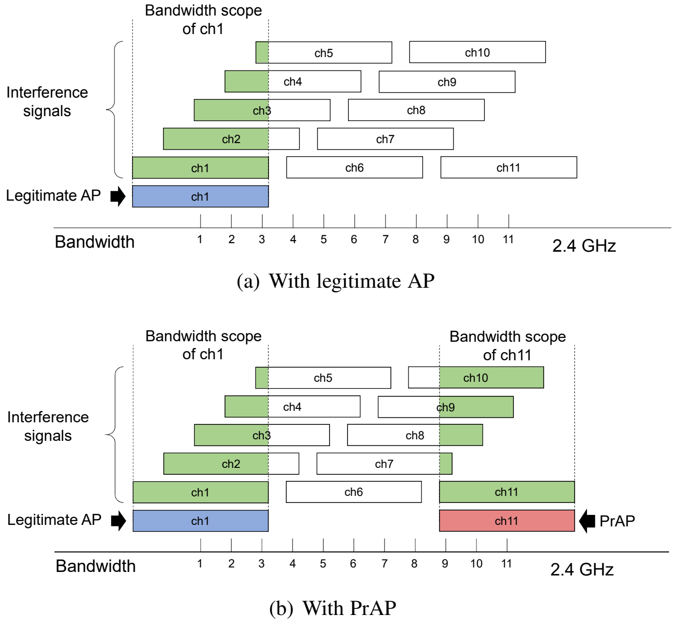
4 Detection Strategy

4.1 The Basic Concept

Our PrAP-Hunter has two wireless interfaces, one that associates itself with a target AP to generate traffic to a receiver (a server listening TCP connection) during detection, while the second interface (interference device) interferes with channel 1 to 11 sequentially with a rest time. Figure 3 illustrates how the proposed method works. The PrAP-Hunter connects to the target AP (ch 11), which relays signals between the legitimate AP (ch 1) and a PrAP-Hunter (ch 11). In the beginning PrAP-Hunter does not know whether the AP is relaying signals or not. When the PrAP-Hunter generates traffic to the receiver, both channels 1 and 11 contribute to the data transmission. From the standpoint of PrAP-Hunter, if obstruction of data transmission is observed at channel 11 when the interference device interferes with channel 1, this is a strong indicator that the target AP is relaying signals wirelessly. When that happens, the connected AP must be a PrAP.

4.2 Channel Interference in 802.11n

As described in the 802.11n standard, the channels used for WLAN are separated by 5MHz in most cases, but have a bandwidth of 20MHz. In other words, each channel shares bandwidth with other adjacent channels. Considering a 20MHz bandwidth channel, there is 17MHz of bandwidth shared between channels 1 and 2, and 2MHz of bandwidth shared between channel 1 and 5 (Table 1). It means when the interference device works on a certain channel, it does not only interfere co-channel, but it also interferes the adjacent channels sharing the bandwidth.



(a) With legitimate AP

(b) With PrAP

Figure 4. Channel interference under IEEE 802.11n

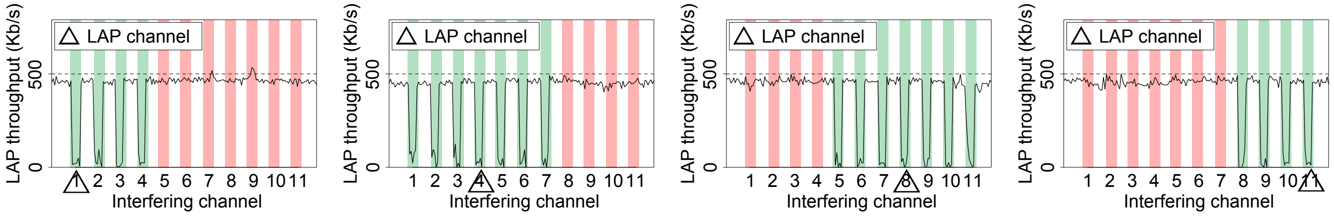
4.3 Advanced Detection Strategy

Figure 4 shows our PrAP detection strategy, considering the wireless bandwidth standpoint. In Figure 4(a), we show a detection scenario where the legitimate AP uses channel 1 and no PrAP exists. We generate traffic through the currently connected AP, while the interference device is transmitting data on channel 1 to 11 with a rest time between each channel interference. When the interference device transmits on channels 1 to 4, the throughput of the legitimate AP at channel 1 is obstructed because of bandwidth sharing as shown in Figure 5(a). Channel 5 also shares a 2MHz bandwidth with channel 1, but 2MHz bandwidth sharing is not enough to interfere substantially. Also, based on both the work in [46] and our experiments, if the PrAP-Hunter and the interference device are located farther than 50cm apart, channel interference caused by 2MHz bandwidth sharing is insignificant. As a result, we obtained Figure 5(a). Throughput degradation for the other channels by the interference are shown in Figure 5, and the channel overlapping shown in Table 1 is confirmed. That is, a channel *ch* is interfered by data transmission over channels from *ch*-3 to *ch*+3 (6 in total, *ch* excluded). For example, throughput on channel 5 would be obstructed by transmission over channels 2,3,4 and 6,7,8. Figure 4(b) shows a detection scenario where a PrAP (ch 11) repeats a signal of legitimate AP (ch 1). If the AP being connected on channel 11 was a legitimate AP, the results of detection should look similar to the results reported in Figure 5(d). However, because we experienced an unexpected throughput degradation on channel 11 as shown in Figure 6(d) when we interfered over channels 1-4 (throughput degradation should have occurred only when interfering over channels 8-11 without a PrAP), we conclude that the connected AP is a PrAP, and it provides wireless connectivity by repeating signals.

To avoid being detected by the obvious throughput degradation by data transmission over unassociated channels, attackers may set a PrAP in an adjacent channel of a legitimate AP, as in Figure 6(a). In this scenario, a PrAP set up at an adjacent channel (ch 1) to a legitimate AP's channel (ch 2). Even in this case, we could observe that transmission over channel 5 also obstructed

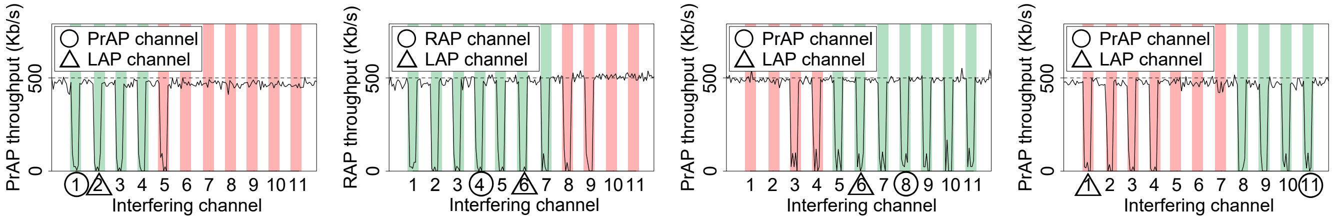
1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing



(a) Channel 1 (b) Channel 4 (c) Channel 8 (d) Channel 11

Figure 5. Cases of only a legitimate AP on various channels. Green bars indicate the overlapped channels with the connecting AP's (here, a legitimate AP) channel affected by interferance, which confirms the channel overlapping model of IEEE802.11n.



(a) (PrAP, legitimate AP) → (1, 2) (b) (PrAP, legitimate AP) → (4, 6) (c) (PrAP, legitimate AP) → (8, 6) (d) (PrAP, legitimate AP) → (11, 1)

Figure 6. Cases of a PrAP existence varying the channel of the PrAP and of the legitimate AP. Red bars indicate the non-overlapped channels. The non-overlapped channels are affected by interference with the channel of the connecting AP (here, a PrAP).

the connected AP's channel (ch 1). The interfering channels, thus, were from 1 to 5 (Figure 6(a)), which were different from 1 to 4 in the legitimate AP case (Figure 5(a)). Therefore, using the interference information, we can infer that the connected AP used not only channel 1 but also channel 2. Simply put, if the number of obstructed channels is more than that of the legitimate AP's only scenario (that is, the number of throughput degradation in Figure 6 is greater than that in Figure 5), there must be a PrAP.

4.4 Implementation Issues

Data transmission. When traffic is generated between the PrAP-Hunter and the traffic receiver, we should guarantee the stability and the speed of the traffic to improve detection accuracy. The location of the traffic receiver is the most important factor in transmission stability, which is affected by traffic volume and dynamic routing paths. Increasing the transmission rate can improve the success rate of detection, but it also increases the workload of the AP, which may affect communication with other users.

Duration and level of interference. If interference with a target channel is too long, it causes nearby devices operating in adjacent channels to experience network delays. This should be avoided by adjusting both the duration and level of interference.

Type of interference message. We need a message that requires a large amount of data, and can guarantee stable interference signals. Also, messages should be broadcast to all devices, because which legitimate AP is used by the PrAP is not known in advance.

Distance between devices. Channel interference is caused by the overlap of adjacent channels [46]. Also, the distance of TX-TX or TX-RX devices affects the interference range of adjacent channels.

5 PrAP-Hunter

We outline a method to derive the degree of channel interference (§5.1), steps to obtain parameters for effective interference (§5.2), and our detection algorithm (§5.3).

5.1 Channel Interference Degree

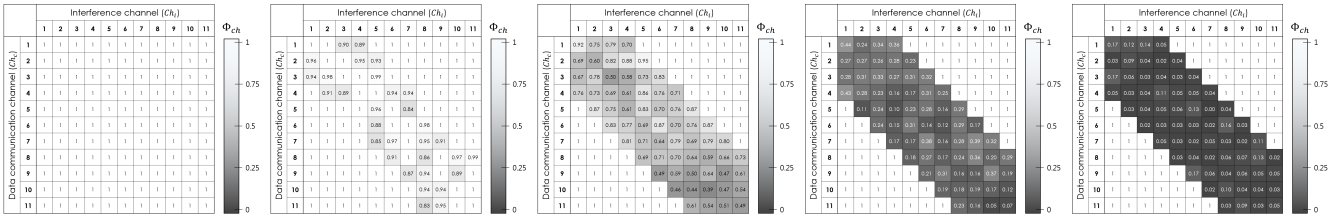
We show how to derive the degree of channel interference, Φ, which is necessary for the operation of PrAP-Hunter. During detection, PrAP-Hunter generates stable traffic to the receiver through the currently connected AP channel (ch_{ap}) and records changes in throughput over regular time intervals. Simultaneously, an interference device generates noise through interference channels (ch) 1 to 11 sequentially. Φ_{ch} is a throughput index of channel ch_{ap} for an interfering channel ch. It has a lower value when transmission over ch does not interfere with the channel ch_{ap}, but a higher value when transmission over ch interfere effectively.

Before each channel interference, the PrAP-Hunter has some rest time for traffic recovery. The PrAP-Hunter calculates the mean throughput during the rest time as ntm_{ch} (normal throughput mean). The PrAP-Hunter also calculates the mean throughput of the AP during the channel interference with ch_{ap} via ch as itm_{ch} (interference throughput mean). Using itm_{ch} and ntm_{ch}, we define the degree of channel interference Φ as

$$\Phi_{ch} = \frac{\mathsf{itm}_{ch}}{\mathsf{ntm}_{ch}}. \tag{1}$$

In this paper, we use a fixed threshold value of 0.5 for Φ_{ch} to determine whether data transmission through the currently connected AP channel (ch_{ap}) is being interfered by noise through the interference channel (ch). From the observation in our experiments, it is hard for Φs to reach values under 0.5 without inducing an intentional channel interference (even when we sent 144 Mbps traffics through the legitimate AP), because the generated traffic volume for detection is quite small. Our system works well in various traffic congestion situations with our threshold as will be shown in §7. If Φ_{ch} is less than 0.5, we determine that the data transmission is being interfered with by noise through ch. Otherwise, we determine that the data transmission is not being interfered with by noise. After interfering with all channels, we obtain Φ values for all of the 11 channels; namely we obtain $\mathsf{R} = \{\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_{10}, \Phi_{11}\}$.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) 30Mbps at 30FPS. (b) 30Mbps at 40FPS. (c) 30Mbps at 50FPS. (d) 30Mbps at 60FPS. (e) 500Kbps at 250FPS.

Figure 7. Examining the performance of modified beacon frame's interference under different conditions. Shaded blocks are affected channels by interference. Darker blocks represent more interference.

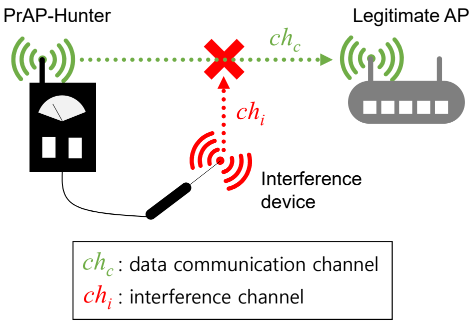

PrAP-Hunter
Legitimate AP
ch_c
ch_i
Interference device
ch_c : data communication channel
ch_i : interference channel

Figure 8. Traffic sent via data communication while beacon is sent via interference channels.

Depending on the being used AP channel (ch_{ap}), we separate a set of interference degrees from R such that

$$\mathsf{B} = \{\Phi_{ch} || \mathsf{ch}_{ap} - ch| > 3, \Phi_{ch} \in \mathsf{R}\}. \quad (2)$$

B includes Φ_{ch}'s, where interference channel (ch) of Φ_{ch} has a channel gap of more than 3 compared with ch_{ap}. That is, we consider only channels (ch) that have no bandwidth shared with ch_{ap}. Why the number "3" is was explained in section 4.3 using Table 1. The reason we care only the channels that is far from ch_{ap} by more than 3 channels is to check whether the throughput on ch_{ap} is obstructed or not by noise transmission over the independent channels from ch_{ap}. If the AP is a legitimate AP, we cannot obtain a Φ_{ch} in the set B that is less than our threshold of 0.5. However, if it is a PrAP, we will obtain at least one Φ_{ch} in the set B that is less than the threshold of 0.5. For evaluating the performance of our method, we collect the minimum Φ_{ch} in the set B after each detection, and denote it by Φ_{min}.

5.2 Efficiency and Impact of Interference

For effective intentional channel interference, we use a modified beacon frame, which will be described in §6.4. The basic function of the beacon frame is to broadcast signals of existence and connection information of the AP to stations. However, beacon frames broadcast too frequently by APs also can be problematic, and may increase the workload of the surrounding devices.

The degree of channel interference depends on the frame rate of the interfering beacon and the data transmission rate of the traffic generator. Figure 7 shows the results of channel interference experiments under various conditions of the experimental setup in Figure 8. In this figure, the number in each block denotes the degree of channel interference, and shaded blocks are the channels affected by interference. Darker blocks represent more interference. As shown in Figure 7(a), at data transmission rate of 30 Mbps over the connected AP channel (Ch_c) and interfering beacon of 30 FPS over the other interfering channels (Ch_i), we did not observe any transmission obstruction, even when interference was sent through channels that had bandwidth shared with the data transmission channels. As shown in Figure 7(b), at 30 Mbps and 40 FPS we observed transmission obstruction with interference signals through a bandwidth-sharing channel. However, degrees of channel interference were not clear enough to conclude that data transmission was obstructed by the interfering beacon frames, because we obtained similar results in some unstable networks. As shown in Figure 7(c), at 30 Mbps and 50 FPS, transmission is obstructed when interference was sent through channels that have bandwidth shared with the data channels. Similar to results in Figure 7(b), we also observed unstable channel interference. Figure 7(a)-(d) show that when interference was sent through channels that had bandwidth sharing, the degree of channel interference was affected by the frame rate of the interference. Figure 7(d) shows that with interference of 60 FPS (frame per second) we could stably interfere with data transmission when an interference signal was sent through bandwidth sharing channels.

It seemed at first to be more advantageous to use a higher transmission rate over Ch_c. However, we noticed that high transmission rate might increase the workload at an AP, and might affect the experience of other users negatively precluding it from real deployment scenarios. Also, we noticed that the data transmission rate significantly depends on the network state and the performance of the AP; the data transmission rate cannot be guaranteed at a high rate. For these reasons, it would be better if we are able to interfere effectively with a lower data transmission rate. Upon various attempts of adjusting the parameters, we obtained the experiment results shown in Figure 7(e), where an interference of 250 FPS effectively worked, even at low data transmission rates (500 Kbps). High-speed beacon transmission also affected other devices that listened to the beacon and increased the error rate of data transmissions. Thus, we need to minimize channel interference time to avoid such side-effects.

5.3 Detection Method

The PrAP detection consists of three algorithms: a PrAP-Hunter, an interference algorithm, and a traffic receiver. We run Algorithm 1 to determine whether the used AP is a PrAP or not.

PrAP-Hunter. Algorithm 1 presents the PrAP-Hunter, consisting of preparation, interference repeating, and traffic analysis phases. The first phase implements the preparation for detection. $\mathsf{Connect}$ first makes association to the target AP where the SSID is SSID_{ap} and obtains channel information ch_{ap}. After association, $\mathsf{Send}_{\mathsf{B}}$, a blocking IO function, builds a TCP/IP connection via ch_{ap} with the traffic receiver ($\mathsf{TrafficReceiver}$) and sends random data ($data$) for Δt time. This is done to check the state of the TCP/IP connection and ensure data transmission rate stability for the

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Algorithm 1 PrAP-Hunter

```
Input: $\text{SSID}_{ap}$
Output: true/false
/* traffic measurement */
$\text{ch}_{ap} \leftarrow \text{Connect}(\text{SSID}_{ap})$
$thput[0] \leftarrow \text{Send}_{\text{B}}(\text{ch}_{ap}, \text{TrafficReceiver}, data, \Delta t)$
for $ch \leftarrow 1$ to $11$ do
    $thput[ch] \leftarrow \text{Send}_{\text{NB}}(\text{ch}_{ap}, \text{TrafficReceiver}, data, \Delta t)$
    $\text{Interfere}_{\text{B}}(\text{InterferenceDevice}, 0.6\Delta t, 0.4\Delta t, ch)$
end for
/* traffic analysis */
for $ch \leftarrow 1$ to $11$ do
    if $|\text{ch}_{ap} - ch| > 3$ then
        /* normal throughput */
        $\text{ntm}_{ch} \leftarrow \text{Mean}(thput[ch][0 \sim 0.6\Delta t])$
        /* throughput under interference */
        $\text{itm}_{ch} \leftarrow \text{Mean}(thput[ch][0.6\Delta t \sim \Delta t])$
        $\Phi_{ch} = \text{itm}_{ch}/\text{ntm}_{ch}$
        if $\Phi_{ch} < 0.5$ then
            return false        /* PrAP */
        end if
    end if
end for
return true        /* Legitimate AP */
```

Algorithm 2 $\text{Interfere}_{\text{B}}$

```
Input: $ch$
Wait($0.6\Delta t$)
SetChannel($ch$)
$t_1 \leftarrow \text{CurrentTime}()$
while $(t_2 - t_1) \leq 0.4\Delta t$ do
    Broadcast($ch, modifiedbeacon$)
    $t_2 \leftarrow \text{CurrentTime}()$
end while
```

next phase. In the second phase, the PrAP-Hunter repeats the channel interference. For each round of interference, Send_{NB}, a non-blocking IO function, sends $data$ to TrafficReceiver for Δt time and records the throughput in $thput[ch]$ during that period of time. The PrAP-Hunter then executes a blocking IO function, the InterferenceDevice rests for the first $0.6\Delta t$. After that, $\text{Interfere}_{\text{B}}$ (Algorithm 2) is executed to interfere with ch for $0.4\Delta t$. Thus, the data structure $thput[ch]$ has the throughput recordings for two parts: the first $0.6\Delta t$ is where there is no interference, and the subsequent $0.4\Delta t$ is for recordings under interference as a result of applying InterferenceDevice on ch.

The last phase is for traffic analysis. The detector calculates the mean normal throughput ntm_{ch} for the period with no interference, and the mean throughput itm_{ch} for the period with interference, and calculates the degree of channel interference as in (1). If Φ_{ch} is less than the threshold of 0.5, we conclude that data transmission via ch_{ap} is obstructed by interference of the other channels, so the AP connected is a PrAP. When any data obstruction at ch_{ap} is observed for the entire period of interference, we conclude that the AP is not a PrAP.

Interference. The algorithm 2 presents the interference procedure. In our method, the interference device does not interfere with a specific AP, but with a specific channel signal, or with all APs using that channel. Therefore, broadcasting frames such as beacon (as opposed to destination-designated frames) fit our purpose. When the PrAP-Hunter starts, the interference device is put in a standby mode waiting for command from the PrAP-Hunter. When the interference device receives channel information ch, it is put into the standby mode for $0.6\Delta t$. After the $0.6\Delta t$ time has passed, it starts broadcasting modified beacon frames for $0.4\Delta t$.

Receiver. The third algorithm, TrafficReceiver, receives data generated by the PrAP-Hunter. TrafficReceiver waits for connections from the PrAP-Hunter. When it receives data from PrAP-Hunter, the receiver discards it to avoid unnecessary waste of resources.

6 Experimental Setup

We implemented PrAP-Hunter in two settings: a high-end hardware (desktop) in a fixed position for analyzing the performance under various traffic scenarios and a mobile PrAP-Hunter was implemented on a relatively low-performance mobile device and is used for analyzing the performance under various locations.

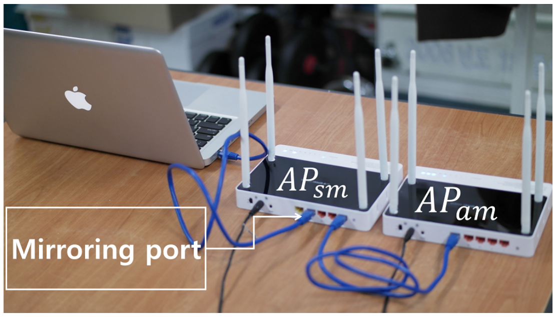


Figure 9. PrAP's Hardware. AP_{sm} is a WLAN router associated with a legitimate AP, and AP_{am} is a WLAN router disguised as a legitimate AP.

6.1 Legitimate AP and PrAP

An EFM ipTIME N8004R is used in our experiments to setup the legitimate AP (in 802.11n mode). In this work, we focus on a PrAP that very quickly repeats the signals of a legitimate AP. The PrAP consists of two WLAN routers (EFM ipTIME N8004R), where one is in the station mode (AP_{sm}) and the other is in an AP mode (AP_{am}). Figure 9 shows the PrAP used in this paper. In this figure, AP_{sm} is responsible for repeating signals to and from the legitimate AP. AP_{am} and AP_{sm} are interconnected using a LAN cable, and AP_{am} is assigned a valid IP from a DHCP server of AP_{sm} with a spoofed SSID and MAC address. Attackers could plug a LAN cable into a port of AP_{am} or AP_{sm} for a port mirroring function that helps data capture much easier. All devices are operated in the IEEE 802.11n mode with MIMO.

6.2 Desktop Detector

The hardware configuration of our desktop PrAP-Hunter is a PC with an Intel Core i5-3570K CPU, 4GB RAM, an ipTIME n500U external wireless card as a traffic generator, and a D-Link DWA-125 external wireless card as an interference device (Figure 10). We implemented our PrAP-Hunter using C# in MonoDevelop (ver.2.8.6.3) supporting a GUI development environment in Linux Ubuntu 12.04 (kernel ver.3.2.0-33-generic). The interference device was implemented in C with the Loss of Radio Connectivity (Lorcon2) library, which is a generic library for injecting 802.11 frames in the MAC layer. Lorcon2 allows modifying 802.11 frames to inject frames through specific channels. As shown in [46], the distance between devices is also an important interference factor. To maintain the same interference conditions, we placed the interference device at the same distance as the PrAP-Hunter, the legitimate AP, and the PrAP, as shown in Figure 11.

6.3 Mobile PrAP-Hunter

Figure 12 shows the hardware configuration of our mobile PrAP-Hunter, which consists of a Google Nexus 5 LG-D821 with a TP-LinkTL-WN722N external wireless card for interference. We used the internal wireless card associated with the mobile device as a traffic generator. For the software, we implemented

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

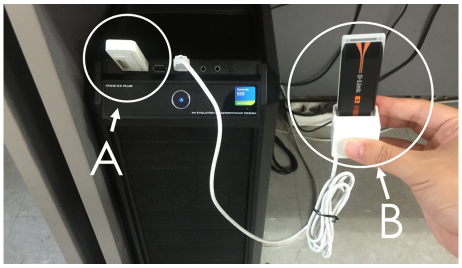


Figure 10. Hardware setting of the desktop PrAP-Hunter. A is an wireless interface that is connected to the target AP and generates traffic, and B is the wireless interface that sends interference signals through 2.4GHz channels.

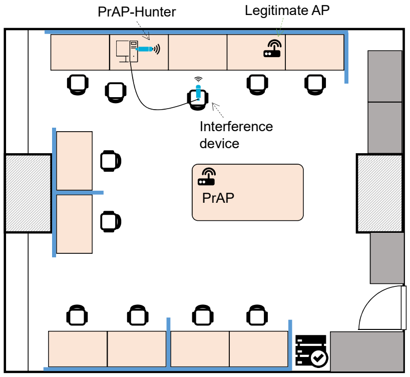


Figure 11. Map of the experiment for the desktop PrAP-Hunter.

the detector with an Android application running Omni-4.4.2-20140513-hammerhead-NIGHTLY with kernel 3.4.0-ElementalX-0.21+. The interference device was implemented in C. The PrAP-Hunter communicates with the interference device through JAVA secure channel (Jsch) library. Cross-compiled Lorcon2 and libpcap libraries were also used for running the interference device.

Experiments are performed in different positions: A and B, as in Figure 13. In position A, the distance between each device was identical to the desktop experiment. With the experiment in position A, we tried to verify the accuracy of the mobile PrAP-Hunter. We chose position B to perform our experiments and to analyze if the position of the PrAP-Hunter affects the results.

6.4 Modified Beacon Frame

In reality, most AP devices construct beacon frames less than 500 bytes in size. However, we needed a beacon frame that contained large amounts of data to stably generate interference signals. Thus, we modified the size of the beacon frame to contain up to 1500 bytes. For sizing up our beacon frame, random information is added in the network data field.

6.5 Time of Detection

A timer was used to record traffic and the interval was set to 0.2 seconds (s). In our experiments, we set Δt to 5s (3s for traffic recovery and 2s for interfering). Furthermore, we interfered with all channels. However, considering that interfering with a channel ch also affects the adjacent *six* channels (from $ch - 3$ to $ch + 3$) owing to the channel overlapping property as shown in Table 1, we do not need to interfere with all channels but with only 2 channels. Thus, we spend 5s at a minimum and 10s at a maximum. When using 5GHz bandwidth, channels do not share bandwidth between each other. Thus, the detection time is the number of channels in the 5GHz multiplied by Δt. Moreover, in the mixed case (i.e., 2.4 GHz and 5.0 GHz), the total detection time is the sum of 2.4 GHz and 5.0 GHz detection times.

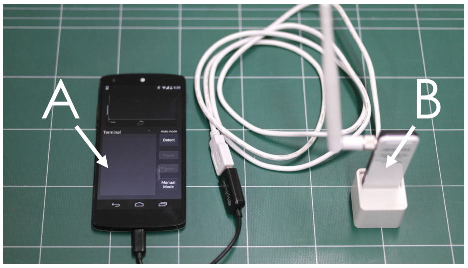


Figure 12. Hardware setting of the mobile PrAP-Hunter.

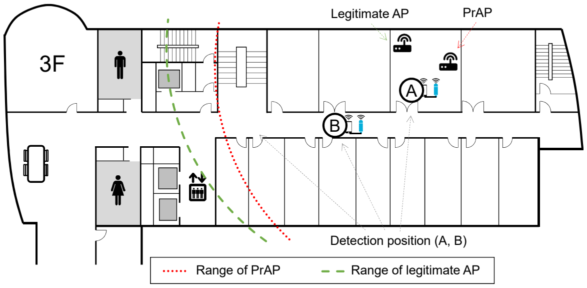


Figure 13. Map of the experiment for the mobile PrAP-Hunter.

7 Evaluation

7.1 Evaluation of PrAP

To evaluate the performance of the PrAP in context, we implemented a time-based detector described by Han *et al.* [3], where they used the round trip times between station and a DNS server and between station and AP to determine whether the used AP is rogue or not. They stated an additional wireless interval led to delay in the round trip time of a DNS query. Their rAP was software-based. We argue that the observed delay was not the result of an additional wireless path, but rather the result of a computational delay caused by the software bridging.

To show that, we performed experiments for Han *et al.*'s algorithm under the rAP and the PrAP. the rAP was configured as in Figure 2(a) and as described in [3] (a software-based rogue AP), and the PrAP was configured as shown in Figure 2(c) (a hardware-based rogue AP), which is the one developed in this work. Figure 14 shows that Han *et al.*'s algorithm could successfully distinguish the legitimate AP and the software-based rAP. However, we also see that the same technique did not work against the hardware-based PrAP (i.e., the mean of Δt is mixed for both the legitimate AP (blue circles) and the PrAP (red crosses). Yang *et al.*'s work also tried to solve the same problem using inter-packet arrival time (IAT) [5]. Although they distinguished one-hop IAT from two-hop IAT to detect the relay attacker, the empirical values to distinguish them were much greater than the theoretical values, which again implies that there is hidden delay caused by the software-based relaying.

7.2 Desktop PrAP-Hunter

Figure 15 summarizes the results of our experiments in an idle and a heavy traffic scenarios. For the idle traffic scenario, experiments were conducted around 3:00 AM at an office space. For the heavy traffic scenario, we used two wireless adapters to generate maximal data rate of 144 Mbps through the legitimate AP, which is

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

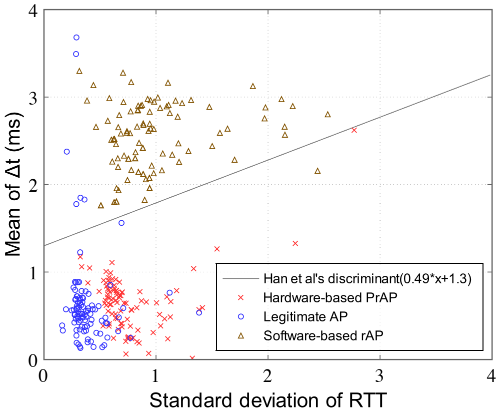


Figure 14. Results of Han *et al.*'s [3] algorithm for two rogue APs, a software-based rAP and a hardware-based PrAP.

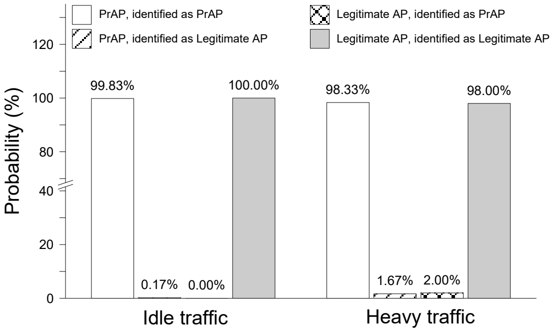
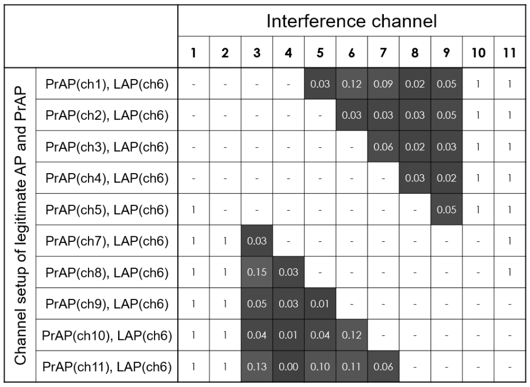


Figure 15. Examining the accuracy of our detection algorithm in different traffic scenarios.

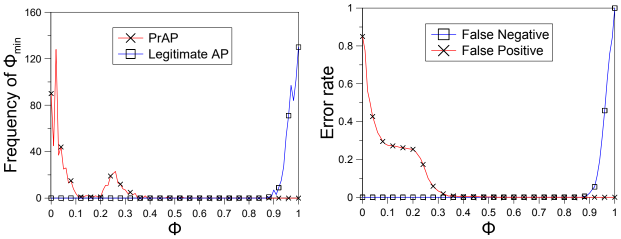
the bandwidth limit of IEEE802.11n with MIMO (two antennas). We conducted experiments for 600 times with a PrAP under idle traffic. As a result, the proposed method only failed one time. We repeated our experiments with a legitimate AP for 600 times, and the proposed method successfully identified the legitimate AP without an error. Similar experiments were conducted in a heavy traffic scenario, and as a result, the method failed 10 times with the PrAP and 12 times with a legitimate AP. In the following, we examine the results of both scenarios in more details.

Results in an Idle Traffic Scenario. In an idle traffic scenario, we examined the proposed method against a PrAP with different channel combinations. Figure 16 shows the results in details. The first column shows the channel setup of the legitimate AP and the PrAP, and the first row lists interference channels (our interference device purposely interferes with the PrAP channel by sending beacons through a legitimate AP's channel.). To detect a PrAP, the PrAP-Hunter connected to the PrAP and it sent data. For simplicity, we only listed Φ_{ch}s of which interference channel ch had a gap of more than 3 channels from the PrAP's channel. As a result, we observed that all the interference channels of which Φ_{ch}s were less than our fixed threshold of 0.5 (from channel 3 to channel 9) shared bandwidth with channel 6 of the legitimate AP. That is, under the existence of a legitimate AP on channel 6, a PrAP will be caught by our algorithm irrespective of what channel the attacker chooses to use. As described in §4, when a PrAP relays traffic between a station and a legitimate AP, the throughput in both channels of the two APs contribute to data transmission. When interference signals are applied to channels that share bandwidth with a legitimate AP, we observe traffic obstruction from the standpoint of the PrAP-Hunter using an independent channel.



		Interference channel										
		1	2	3	4	5	6	7	8	9	10	11
Channel setup of legitimate AP and PrAP	PrAP(ch1), LAP(ch6)	-	-	-	-	0.03	0.12	0.09	0.02	0.05	1	1
	PrAP(ch2), LAP(ch6)	-	-	-	-	-	0.03	0.03	0.03	0.05	1	1
	PrAP(ch3), LAP(ch6)	-	-	-	-	-	-	0.06	0.02	0.03	1	1
	PrAP(ch4), LAP(ch6)	-	-	-	-	-	-	-	0.03	0.02	1	1
	PrAP(ch5), LAP(ch6)	1	-	-	-	-	-	-	-	0.05	1	1
	PrAP(ch7), LAP(ch6)	1	1	0.03	-	-	-	-	-	-	-	1
	PrAP(ch8), LAP(ch6)	1	1	0.15	0.03	-	-	-	-	-	-	1
	PrAP(ch9), LAP(ch6)	1	1	0.05	0.03	0.01	-	-	-	-	-	-
	PrAP(ch10), LAP(ch6)	1	1	0.04	0.01	0.04	0.12	-	-	-	-	-
	PrAP(ch11), LAP(ch6)	1	1	0.13	0.00	0.10	0.11	0.06	-	-	-	-

Figure 16. Features of Φ_{ch} shown under an idle traffic scenario of the desktop PrAP-Hunter (500Kbps, 250FPS). RAP is the currently-connected AP, and it is relaying signals between a PrAP-Hunter and a legitimate AP (LAP).



(a) Distribution of Φ_{min}s (b) Error rate against Φ

Figure 17. (a) Desktop PrAP-Hunter: distribution of Φ_{min}s; idle traffic. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) CDF of false negative and false positive rates against Φ.

We collected all instances of Φ_{min} in each detection trial to analyze the distribution in idle traffic experiments. As shown in Figure 17(a), when we tested our algorithm with PrAP, most of the Φ_{min}s in each detection trial were less than 0.4. With a legitimate AP, all Φ_{min}s in each detection trial were greater than 0.87.

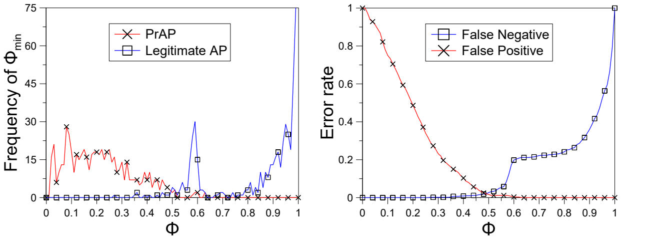
Figure 17(b) shows the legitimate AP's and PrAP's detection error rate against Φ. The detection threshold was between 0.54 and 0.87, which kept both false positive and false negative rates at 0%. Even though we used a fixed detection threshold at 0.5 to distinguish legitimate APs and PrAPs, we could obtain a false positive rate of 0% and a low false negative rate at less than 1%.

Results in a Heavy Traffic Scenario. Results in a heavy traffic scenario are almost identical to those in the idle scenario. Distribution in a heavy traffic case in Figure 18(a) looks more noisy than that in the idle case in Figure 17(a). However, as shown in Figure 18(a), for the PrAP, most Φ_{min}'s in each detection attempt were less than the fixed detection threshold of 0.5. With a legitimate AP, most Φ_{min}'s in each detection attempt were greater than 0.5. Figure 18(b) shows the legitimate AP and PrAP detection error rate against Φ. We observe that a detection threshold of 0.49–0.50 could keep both false positive and false negative rates less than 2%. In this paper, we used a fixed detection threshold at 0.5 to distinguish legitimate APs and PrAPs, which produced a sum of false positive and false negative rate of less than 3.67%.

7.3 Mobile PrAP-Hunter

We performed our experiments in an idle traffic scenario with the mobile PrAP-Hunter, since we only wanted to know whether our method could work well when the PrAP-Hunter is placed far from both the legitimate AP and the PrAP (c.f. §8 for the performance in a heavy traffic scenario). For that reason, we performed our

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) Distribution of Φ_{min}s (b) Error rate against Φ

Figure 18. (a) Desktop PrAP-Hunter: distribution of Φ_{min}s; heavy traffic. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) CDF of false negative and false positive rates against Φ.

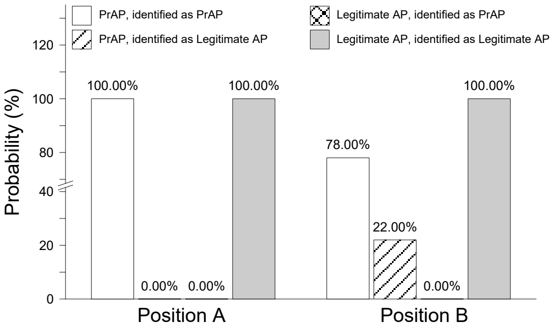
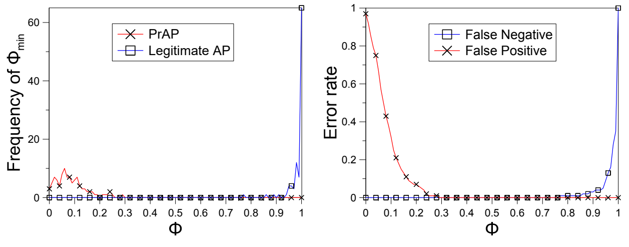


Figure 19. A plot examining the accuracy (measured by the true positive, true negative, false positive and false negative) of our detection algorithm at different positions.

experiments in two different positions: A and B, as shown in Figure 13. Figure 19 summarizes the results of our experiments using the mobile PrAP-Hunter. In each position, we examined the proposed method against both legitimate AP and PrAP 100 times, respectively. As a result, experiments in position A showed 100% success rate in detecting both legitimate AP and PrAP. In position B, the PrAP-Hunter failed 22 times against the PrAP and never failed against the legitimate AP.

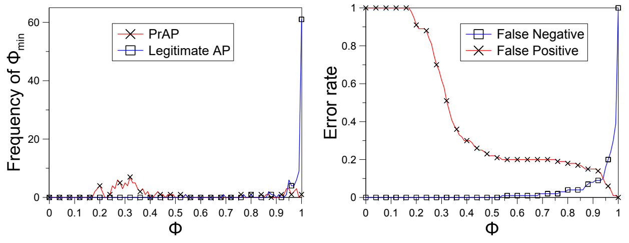
Results in Position A. The experiment setting was same as in the desktop PrAP-Hunter experiment. As shown in Figure 20(a), the distribution of the Φ_{min} was similar to the results shown in the idle traffic scenario of the desktop PrAP-Hunter experiments. Legitimate APs and PrAPs could clearly be distinguished, because all Φ_{min}s for each PrAP detection were less than 0.3, and for legitimate AP detection, they were greater than 0.85. Figure 20(b) shows the legitimate AP and the PrAP detection error rate against Φ. As shown in the figure, we can keep both false positive and false negative rates at 0% when we set the detection threshold between 0.3 and 0.78. Thus, when we use a fixed detection threshold at 0.5, our PrAP-Hunter produced a 100% success rate in both legitimate AP and PrAP detections.

Results in Position B. We repeated detection experiments 100 times for each legitimate AP and PrAP in position B, where the distance between them is 10 meters. Figure 21(a) shows the distribution of Φ_{min}, where the PrAP-Hunter was able to maintain a stable data transmission with the used AP. Thus, when we tested the proposed method with a legitimate AP, most of the Φ_{min} values were greater than 0.8. With the PrAP, most Φ_{min} values in each trial were less than 0.5, and greater than 0.8 only in a few cases; that happened only when the channel gap between the legitimate AP and the PrAP was only 1. For example, assume that a legitimate AP used channel 6 and a PrAP used channel 5. We should interfere only with channel 6, the legitimate AP's channel, but not the PrAP's channel (channel 5) to observe the traffic obstruction by the interference. To do so, channel 9 would be the best choice. When the interference device works on channel 9, it would interfere with the legitimate AP's channel (channel 6) via overlapping channels 6, 7, and 8 successfully, but not the PrAP's (channel 5). Unfortunately, signals of the interference device are attenuated significantly due to the distance. Thus, the number of channels affected by the interference device was only 5 (channels 7, 8, 9, 10, and 11. Not 7 channels as we expected in our detection strategy). That is, only channels 7 and 8 (but not channel 6) were affected, so beacon transmission over channel 9 did not successfully interfere with the legitimate AP's channel. This exceptional case happens only when: (1) the PrAP-Hunter is far both from legitimate AP and PrAP, and (2) the service channel gap is 1. However, we can break condition (1) by moving our PrAP-Hunter closer to an AP of interest using proper SNR values.



(a) Distribution of Φ_{min} (b) Error rate against Φ

Figure 20. Results. (a) Mobile PrAP-Hunter: the distribution of Φ_{min} at position A in our experimental setup. The detection trials were repeated 100 times for the rogue and legitimate AP measurements, respectively. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) The CDF of the false negative and false positive rates against various values of Φ.



(a) Distribution Φ_{min} (b) Error rate against the Φ

Figure 21. Results. (a) Mobile PrAP-Hunter: the distribution of Φ_{min} at position B, where detection trials were repeated 100 times for both the rogue and legitimate AP measurements, respectively. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) The CDF of the false negative and false positive rates against various values of Φ.

Remark on detection position. There was one extreme case where the PrAP-Hunter was out of the range of the legitimate AP, but within the range of the PrAP. Even in this case, the PrAP-Hunter still could interfere with the communicating channel between the legitimate AP and the PrAP, and it successfully detected the PrAP, because the PrAP's network interface was still within the PrAP-Hunter's range.

In summary, we conclude that PrAP-Hunter's position affects detection performance: a larger distance between PrAP-Hunter and APs caused a higher error rate. Generally, since a station will select an AP with the highest SNR, a PrAP should be located close to the station to allow connection. Also, a network administrator using our PrAP-Hunter can easily find the location close to an AP of interest using SNR values, and scenarios shown in position B can be easily avoided. Figure 22 summarizes our experiments.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

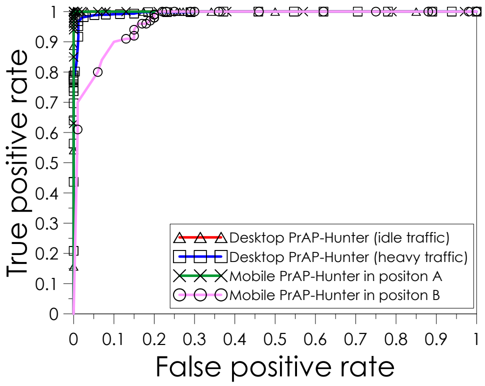


Figure 22. The Receiver Operating Characteristic (ROC) curve capturing the trade-off between the false and true positive rates, and various experimenting settings for both the desktop and mobile versions of PrAP-Hunter. The results show that one can choose an optimal threshold to achieve a high true positive without sacrificing the low false positive rate in most settings.

8 Detection In the Wild

To illustrate PrAP-Hunter in the real-world, we conducted experiments at coffee shops upon obtaining the store's permission and an approval from our institutional review board (IRB) assuring that our experiments are in no way going to harm users.

8.1 Hide-and-Seek Game

We designed a "hide-and-seek" game to show how our PrAP-Hunter performs in real world (10 different coffee shops). Below, we describe this game; settings, detection strategy and results.

Settings. For this game, we had two players: attacker (hider) and PrAP-Hunter (seeker). We designed and developed our hardware PrAP so that it was easily deployed in real world: it only needed a power source for operation with all parameters pre-defined and set. For our experiments, the attacker may (or may not) decide to deploy a PrAP in the tested environment. If he decides to deploy a PrAP, the PrAP was turned on and its position was determined by the attacker. For more realistic experiments, the location of the PrAP was chosen randomly. PrAP-Hunter (the defender) knew the location of the legitimate AP, since it was visible to users as well as PrAP-Hunter. However, PrAP-Hunter did not know the location of PrAP nor whether a PrAP was turned on or off. The PrAP-Hunter was assumed to automatically connect to the PrAP when it had the highest power signal in the deployment environment. We noticed that this assumption was reasonable: in all the stores where we ran our game, the default Wi-Fi manager did not allow choosing an SSID working on a specific channel, but rather automatically connected to the AP with the highest power.

Strategy. First, the PrAP-Hunter finds the position of the legitimate AP, which is visible and often located by the cashier as shown in Figure 24(a). Then, the PrAP-Hunter chooses a Wi-Fi connection position, and our choice of this position must ensure that the PrAP has a stronger signal than the legitimate AP's, so that a legitimate user may connect to the PrAP automatically. Accordingly, the Wi-Fi connection position must be far from the visible legitimate AP. Once connected, we start the detection phase.

Results. Based on the settings and strategy described above, the two players execute the game: one player hides the PrAP and the other tries to find it. The PrAP is turned either on or off by the hider, but the choice is not known to the seeker (PrAP-Hunter). After all set up, the seeker comes into the store, and tries to find whether a PrAP exists or not using our PrAP-Hunter. In the experiment, the detection rate was 100%, that is, the seeker correctly found 3 PrAPs and 7 legitimate APs at 10 different stores, which corresponds to the actual deployment of PrAPs. About 20-50 SSIDs were found in each store, and the experiments were conducted in the afternoon. The results are in Figure 23.

8.2 Understanding the Effect of Distance

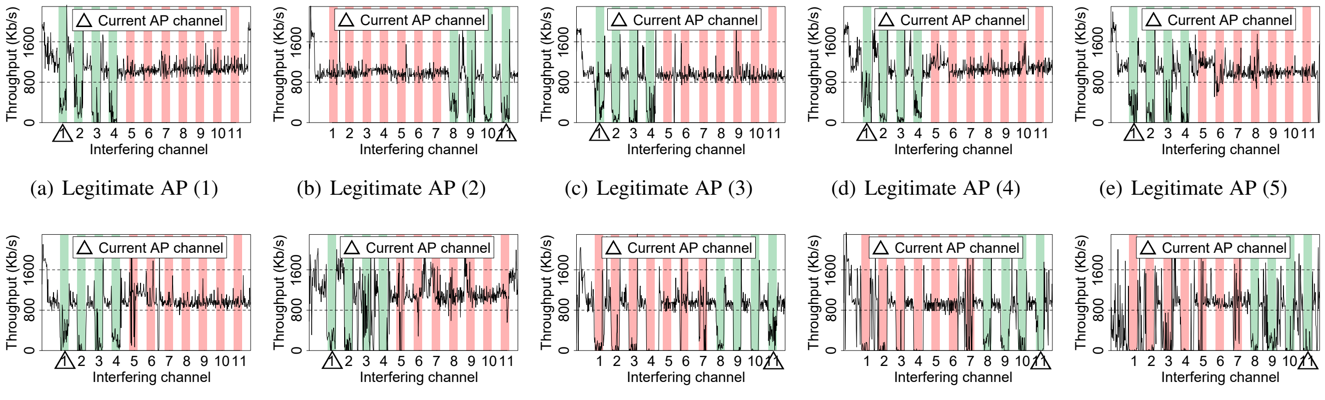
To understand the effect of distance, we ran an experiment at a coffee shop, with the map shown in Figure 24(a). This cafe provided free WLAN with the same SSID at channel 1, 6, 11 and outlet services for customers. With these resources, we set up a PrAP as described in this paper. The PrAP relayed traffic of a legitimate AP serving on channel 1, and operated on channel 11 with the same SSID as a trusted AP. When we tested a legitimate AP using our PrAP-Hunter, we obtained results which were similar to Figure 5(a). If the AP on channel 11 was a legitimate AP, we should obtain results which were similar to the Figure 5(d). However, we obtained results which were similar to Figure 6(d), which means our PrAP-Hunter identified correctly the PrAP.

We performed our experiments in three different positions of the cafe. In position A, the PrAP-Hunter was close to the PrAP. In position B, the PrAP-Hunter had the same distance with both the legitimate AP and the PrAP. In position C, the PrAP-Hunter was close to the legitimate AP. Figure 24(b)-(d) shows results of PrAP detections in three positions. In the experiment, the PrAP-Hunter could not maintain a stable data transmission rate with the rogue AP. The reasons were as follows. First, we placed our traffic receiver in the intranet of our campus. When the PrAP-Hunter sent data from Internet to the intranet, the Internet traffic and dynamic routing paths led to unstable traffic transmission. Second, we performed our experiments at the peak time at the cafe, where another potential reason could be the high level of AP workload. Finally, we placed our PrAP in a backpack to hide it from people, so two wireless interfaces of the PrAP were placed too closely, which caused interference with each other in a small backpack, even though they used different channels [46]. As shown in Figure 24, although PrAP-Hunter showed unstable data transmission, we still could obtain good results that had similar features to Figure 6(d); the proposed method successfully found the existence of a PrAP even in various real world scenarios.

9 Discussion

A sophisticated rogue AP. We analyzed the security against a more sophisticated rogue AP that intentionally reduces the bandwidth of forwarding link when the PrAP-Hunter generates traffic, and increases the bandwidth when PrAP-Hunter generates interference. The derived degree of channel interference might be above the threshold, so the rogue AP might be able to avoid the detection in this case. We break down the sophisticated attacker's control of bandwidth into two cases: one is to increase the bandwidth while interfering, and the other is to decrease it while idle. For the increment of bandwidth, it does not affect our detection algorithm, because our interference is effective on the wireless channel. That is, even when the attacker increases the bandwidth while interfering by giving higher priority to the flow, it cannot increase the throughput for the given wireless channel. When decreasing the bandwidth, we note that our constant-rate traffic is already very low (only 500 Kbps in our experiments). Thus, even the lower transmission rate caused by the attacker

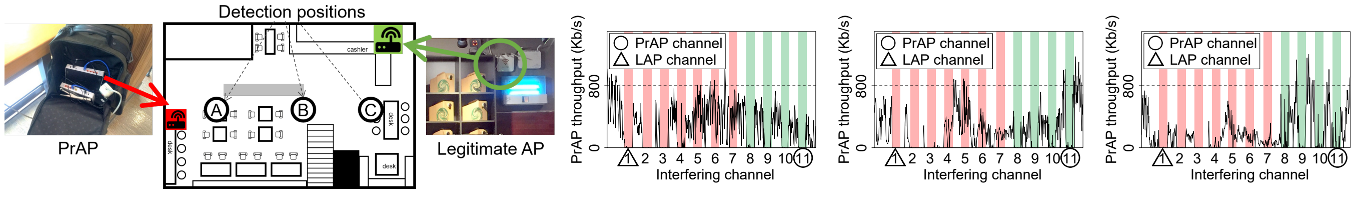
1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) Legitimate AP (1) (b) Legitimate AP (2) (c) Legitimate AP (3) (d) Legitimate AP (4) (e) Legitimate AP (5)

(f) Legitimate AP (6) (g) Legitimate AP (7) (h) PrAP (1) (i) PrAP (2) (j) PrAP (3)

Figure 23. Results of the hide-and-seek game at 10 coffee shops.



(a) Map of the cafe for the experiment. (b) Position A (c) Position B (d) Position C

Figure 24. Legitimate AP serving in channel 1. Rogue AP relayed the Internet service of the legitimate AP and served as a trusted AP (using a cafe's wireless network SSID) in channel 11. In position A, the PrAP-Hunter was close to the PrAP. In B, the PrAP-Hunter had the same distance with both the legitimate AP and the PrAP. In C, the PrAP-Hunter was close to the legitimate AP.

can be a good indicator of the rogue APs existence. Finally, we note that to mount this attack, further investigation is required on whether an attacker is able to identify our detector's traffic or not to frustrate our detection algorithm: considering that the traffic generation and detection can be done in a random interval and duration, they will be mixed with normal users traffic.

Operation mode. Our design of PrAP-Hunter is generic, and is not limited to an after-fact deployment. When a rogue AP is detected using this algorithm, that same rogue AP might has already been used to successfully launch an array of attacks on innocent users connecting to it. To provide preventive countermeasure, we could use our system proactively and periodically to detect malicious and rogue access points as soon as they are deployed. We can set up our system to detect rogue APs in multiple fixed locations and to have them run the detection algorithm periodically, which will give us a high chance to detect rogue APs before they mount the attack.

Detection period. The more frequently the detector is operated, the worse the experience of the legitimate users would be due to interference, although the faster the detection is. This trade-off is a clear limitation of our approach. To balance this trade-off, our approach can be deployed over limited periods of time to detect the rogue APs. Furthermore, in order to address scenarios where the adversary would learn the operation cycles of our detector and try to avoid them by on/off operation, we can also envision that our system would operate by randomly hopping in the time domain for its operation, to be unpredictable.

Interference in 40MHz channels and 802.11ac. For wider channels in 2.4 GHz such as 40MHz, we note that 20MHz channel is most common, and 40MHz is hardly observed because the number of orthogonal channels is too small. Detecting in the 5GHz channels (as in IEEE802.11ac) is much easier with our advanced detection strategy, because channels do not share bandwidth between each other. Similarly, detecting a PrAP relaying 2.4GHz and 5GHz is easy with our strategy. We can interfere with one of the channels (either 2.4GHz or 5GHz channel) while sending data with the other channel to see whether it has two wireless channels or not.

3G/LTE channel. One possible system model scenario in which our attack would operate is a 3G/LTE channel used for relaying. In particular, one may assume that the attackers use such a 3G/LTE network to provide connectivity, by relying the rAP traffic of legitimate users, thus virtually violating the underlying assumption of our detector. In reality, however, in most of the case such an approach for relying traffic would still also be software-bridged, which would make detection even easier based on the time feature. Certainly, one can also perform a 3G/LTE-WiFi hardware bridge to avoid packet delay and eliminate the time feature used for the time-based detection. However, as a defense, one can also employ the same approach of WiFi jamming, utilized in our work for hardware-based rAP detection, but at the cellular network [47] to obtain similar detection results. Testing such a scenario experimentally is an orthogonal contribution to our work, and we will pursue that as a future study.

10 Conclusion

We introduced a PrAP that can evade the most widely advocated and used time-based detection techniques. We showed that while time-based techniques were indeed suitable for software-based rAP detection, they were obsolete against our new PrAP. Using various experiments, we showed the feasibility of our PrAP. To defend against its threat, we developed a new mechanism that used channel interference for PrAP detection. Our mechanism is capable of detecting hardware-based PrAPs, as demonstrated by various experimental scenarios and two deployment setups.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

Acknowledgments. This work was supported by the Global Research Lab. (GRL) Program of the National Research Foundation (NRF) funded by Ministry of Science, ICT (Information and Communication Technologies) and Future Planning (NRF-2016K1A1A2912757).

References

[1] R. Jang, J. Kang, A. Mohaisen, and D. Nyang, "Rogue access point detector using characteristics of channel overlapping in 802.11n," in *Proc. of IEEE International Conference on Distributed Computing Systems, ICDCS*, 2017, pp. 2515–2520.

[2] S. Brenza, A. Pawlowski, and C. Pöpper, "A practical investigation of identity theft vulnerabilities in eduroam," in *Proc. of ACM Conference on Security & Privacy in Wireless and Mobile Networks, S&P*, 2015, pp. 14:1–14:11.

[3] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A timing-based scheme for rogue AP detection," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 11, pp. 1912–1925, 2011.

[4] H. Gonzales, K. S. Bauer, J. Lindqvist, D. McCoy, and D. C. Sicker, "Practical defenses for evil twin attacks in 802.11," in *Proc. of IEEE Global Communications Conference, GLOBECOM*, 2010, pp. 1–6.

[5] C. Yang, Y. Song, and G. Gu, "Active user-side evil twin access point detection using statistical techniques," *IEEE Trans. Information Forensics and Security*, vol. 7, no. 5, pp. 1638–1651, 2012.

[6] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engel, "Undesired relatives: protection mechanisms against the evil twin attack in IEEE 802.11," in *Proc. of ACM Symposium on QoS and Security for Wireless and Mobile Networks, Q2SWinet*, 2014, pp. 87–94.

[7] ——, "Hacker's toolbox: Detecting software-based 802.11 evil twin access points," in *Proc. of IEEE Annual Consumer Communications and Networking Conference, CCNC*, 2015, pp. 225–232.

[8] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Pironti, and P. Strub, "Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS," in *Proc. of IEEE Symposium on Security and Privacy, S&P*, 2014, pp. 98–113.

[9] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "Sok: Security and privacy in implantable medical devices and body area networks," in *Proc. of IEEE Symposium on Security and Privacy, S&P*, 2014, pp. 524–539.

[10] A. M. Bates, J. Pletcher, T. Nichols, B. Hollembaek, D. Tian, K. R. B. Butler, and A. Alkhelaifi, "Securing SSL certificate verification through dynamic linking," in *Proc. of ACM SIGSAC Conference on Computer and Communications Security, SIGSAC*, 2014, pp. 394–405.

[11] N. Cheng, "Take precautions on public Wi-Fi," The Nation, August 2016.

[12] L. Constantin, "This tool can alert you about evil twin access points in the area," InfoWorld, April 2015.

[13] R. Gery, "Beware the 'evil' Wi-Fi networks that turn your phone into a brick: Hackers can hijack systems to remotely attack your handset," Dailymail, April 2015.

[14] A. Baxter, "How to stop hackers from stealing your information on public Wi-Fi," The Next Web, June 2015.

[15] E. Tuvey, "The dangers of using public Wi-Fi hotspots," betnews, March 2016.

[16] P. Cooper, "White hat hackers steal data from london wi-fi users in "evil twin" attack," Security News, November 2013.

[17] R. A. Beyah and A. Venkataraman, "Rogue-access-point detection: Challenges, solutions, and future directions," *IEEE Security & Privacy*, vol. 9, no. 5, pp. 56–61, 2011.

[18] N. Cheng, "Take precautions on public Wi-Fi," The Star, August 2016.

[19] A. Adya, P. Bahl, R. Chandra, and L. Qiu, "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks," in *Proc. of ACM Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2004, pp. 30–44.

[20] S. Nikbakhsh, A. B. A. Manaf, M. Zamani, and M. Janbeglou, "A novel approach for rogue access point detection on the client-side," in *Proc. of International Conference on Advanced Information Networking and Applications Workshops, WAINA*, 2012, pp. 684–687.

[21] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Enhancing the security of corporate Wi-Fi ntworks using DAIR," in *Proc. of ACM International Conference on Mobile Systems, Applications, and Services, MobiSys*, 2006, pp. 1–14.

[22] R. Chandra, J. Padhye, A. Wolman, and B. Zill, "A location-based management system for enterprise wireless LANs," in *Proc. of USENIX Symposium on Networked Systems Design and Implementation NSDI*, 2007.

[23] ——, "A location-based management system for enterprise wireless LANs," 2007.

[24] R. A. Beyah, S. Kangude, G. Yu, B. Strickland, and J. A. Copeland, "Rogue access point detection using temporal traffic characteristics," in *Proc. of IEEE Global Telecommunications Conference, GLOBECOM*, 2004, pp. 2271–2275.

[25] L. Watkins, R. A. Beyah, and C. L. Corbett, "A passive approach to rogue access point detection," in *Proc. of IEEE Global Communications Conference, GLOBECOM*, 2007, pp. 355–360.

[26] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. of ACM Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2008.

[27] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proc. of ACM Workshop on Wireless Security*, 2006, pp. 43–52.

[28] F. Lanze, A. Panchenko, B. Braatz, and T. Engel, "Letting the puss in boots sweat: detecting fake access points using dependency of clock skews on temperature," in *Proc. of ACM Symposium on Information, Computer and Communications Security, ASIACCS*, 2014, pp. 3–14.

[29] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 MAC layer spoofing using received signal strength," in *Proc. of the 27th Conference on Computer Communications, INFOCOM*, 2008.

[30] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *IEEE Trans. Mob. Comput.*, vol. 9, no. 3, pp. 449–462, 2010.

[31] W. Wei, S. Jaiswal, J. F. Kurose, and D. F. Towsley, "Identifying 802.11 traffic from passive measurements using iterative bayesian inference," in *Proc. of IEEE Conference on Computer Communications, INFOCOM*, 2006.

[32] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. F. Towsley, "Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs," in *Proc. of ACM SIGCOMM Internet Measurement Conference, IMC*, 2007, pp. 365–378.

[33] G. Qu and M. M. Nefcy, "Rapid: An indirect rogue access points detection system," in *Proc. of IEEE International Performance Computing and Communications Conference, IPCCC*, 2010, pp. 9–16.

[34] A. Venkataraman and R. A. Beyah, "Rogue access point detection using innate characteristics of the 802.11 MAC," in *Proc. of EAI International Conference Security and Privacy in Communication Networks, SecureComm*, 2009, pp. 394–416.

[35] C. D. Mano, A. Blaich, Q. Liao, Y. Jiang, D. A. Cieslak, D. Salyers, and A. Striegel, "RIPPS: rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 2, pp. 2:1–2:23, 2008.

[36] —, "Madwifi project," http://madwifi-project.org, November 2015.

[37] ——, "Aircrack-ng," http://www.aircrack-ng.org, November 2015.

[38] ——, "The karma software patch for access points," http://digi.ninja/karma, November 2015.

[39] T. Kindberg, J. Mitchell, J. Grimmett, C. Bevan, and E. O'Neill, "Authenticating public wireless networks with physical evidence," in *Proc. of the 5th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob*, 2009, pp. 394–399.

[40] V. Roth, W. Polak, E. G. Rieffel, and T. Turner, "Simple and effective defense against evil twin access points," in *Proc. of ACM Conference on Wireless Network Security, WISEC*, 2008, pp. 220–235.

[41] K. S. Bauer, H. Gonzales, and D. McCoy, "Mitigating evil twin attacks in 802.11," in *Proc. of IEEE International Performance, Computing and Communications Conference, IPCCC*, 2008, pp. 513–516.

[42] H. Gonzales, K. S. Bauer, J. Lindqvist, D. McCoy, and D. C. Sicker, "Practical defenses for evil twin attacks in 802.11," in *Proc. of IEEE Global Communications Conference, 2010. GLOBECOM*, 2010, pp. 1–6.

[43] J. Pang, B. Greenstein, M. Kaminsky, D. McCoy, and S. Seshan, "Wifi-reports: Improving wireless network selection with collaboration," *IEEE Trans. Mob. Comput.*, vol. 9, no. 12, pp. 1713–1731, 2010.

[44] Q. Wang, P. Xu, K. Ren, and X. Li, "Towards optimal adaptive UFH-based anti-jamming wireless communication," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 16–30, 2012.

[45] E. G. Villegas, E. López-Aguilera, R. Vidal, and J. Paradells, "Effect of adjacent-channel interference in IEEE 802.11 WLANs," in *Proc. of IEEE International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications, CROWNCOM*, 2007, pp. 118–125.

[46] A. Zubow and R. Sombrutzki, "Adjacent channel interference in IEEE 802.11n," in *Proc. of IEEE Wireless Communications and Networking Conference, WCNC*, 2012, pp. 1163–1168.

[47] M. Lichtman, J. H. Reed, T. C. Clancy, and M. Norton, "Vulnerability of LTE to hostile interference," in *Proc. of IEEE Global Conference on Signal and Information Processing, GlobalSIP*, 2013, pp. 285–288.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

Catch Me If You Can: Rogue Access Point Detection Using Intentional Channel Interference

Rhongho Jang, Jeonil Kang, Aziz Mohaisen, *Senior Member, IEEE,* DaeHun Nyang, *Member, IEEE.*

Abstract—In this paper, we introduce a powerful hardware-based rogue access point (PrAP), which can relay back and forth traffic between a legitimate AP and a wireless station, and act as a man-in-the-middle attacker. Our PrAP is built of two dedicated wireless routers interconnected physically, and can relay traffic rapidly between a station and a legitimate AP. Through experiments, we demonstrate that the state-of-the-art time-based rogue AP (rAP) detectors cannot detect our PrAP, although perhaps effective against software-based rAP. In demonstrating that, we unveil new insight into fundamentals of time-based detectors for software-based rAPs and their operation: such techniques are only capable of detecting rAPs due to the speed of wireless AP bridging. To address the threat of such PrAPs, we propose a new tool for network administrators, a PrAP-Hunter based on intentional channel interference. Our PrAP-Hunter is highly accurate, even under heavy traffic scenarios. Using a high-performance (desktop) and low-performance (mobile phone) experimental setups of our PrAP-Hunter in various deployment scenarios, we demonstrate close to 100% of detection rate, compared to 60% detection rate by the state-of-the-art. We show that our PrAP-Hunter is fast (takes 5-10 seconds), does not require any prior knowledge, and can be deployed in the wild by real world experiments at 10 coffee shops.

Index Terms—Intrusion detection, Wireless LAN, Rogue AP, channel interference, IEEE 802.11n.

1 Introduction

The Wireless Local Area Network (WLAN) as a technology is popular in part for supporting mobility, a feature that makes WLAN deeply integrated in many essential applications to facilitate an easy Internet access at public spaces such as restaurants, cafes, and public libraries. Due to the mobility features in WLAN, their easy setups, and the lax use policies in many deployment scenarios, various security threats have emerged. For example, while WLAN allows users to easily set a new wireless Access Point (AP) up using off-the-shelf WLAN hardware and their electronic gadgets (e.g., laptop computer or smartphone), that same feature also allows an adversary to set up a rogue AP (rAP), for potentially attacking benign users. Even worse, many WLAN users are unaware of the security dangers associated with wireless access, especially when connecting to APs in public places.

With many public spaces, including shopping malls, restaurants, and public transit systems, providing WLAN services and power outlets for customers, an adversary equipped with a laptop and an additional network interface can easily create a *persistent* rAP to eavesdrop on, intercept, or even modify communications between users and the Internet. An adversary capable of creating such rAP can use it to launch a large array of attacks on innocent users connecting to it. For example, the attacker can eavesdrop on the exchange of sensitive information such as identity credentials, password, and bank account by observing relayed packets as shown by Brenza *et al.* [2]. The attacker can also mount an active attack by rewriting DNS queries and response to lead users to phishing websites. The attacker can even infect the user's device with a malicious software (malware) by reflecting malicious contents in response to the user's browsing requests.

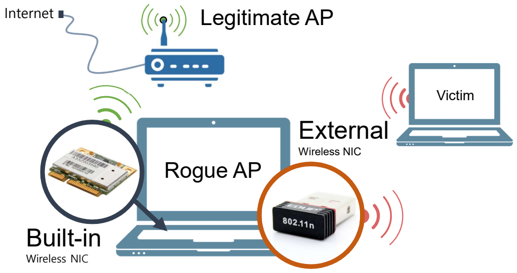


Figure 1. General rogue Access Point (rAP) setup using a laptop with a built-in and an external wireless interfaces.

Indeed, various recent research studies have pointed out and experimentally demonstrated this security issue as a threat [3], [4], [5], [6], [7]. Also, several research results showed that threats caused by the rogue AP were quite practical even using TLS and SSL [8], [9], [10]. Moreover, various recent media reports (from 2013 to 2016) have detailed real attack incidents using rAPs, and stressed various security, privacy, and public safety implications [11], [12], [13], [14], [15], [16]. For example, in late 2013, researchers from Trend Micro™ conducted a rAP experiment in the city of London, where they found that: 1) users deliberately connected to such AP for various online activities, and 2) users liberally exposed their information to the attacker, including login credentials, transactions, and other sensitive information [16].

As in the literature, a rAP is typically created using two wireless interfaces, where the first interface is built-in while the second one is external as shown in Figure 1 as shown in [17]. The built-in interface (e.g., inside a laptop) is operated in the station mode and used for connecting to a legitimate AP, while the external interface is operated in the service mode and used by users as trusted AP. The two interfaces forward packets to each other and relay Internet services on behalf of the user (victim). In

D.H. Nyang is the corresponding author. R. Jang, J. Kang and D.H. Nyang are with the Department of Computer Engineering, Inha University, Incheon 22212, Korea. R. Jang (second affiliation) and A. Mohaisen are with the Department of Computer Science at the University of Central Florida, FL, USA. (e-mail: r.h.jang@knights.ucf.edu, dreamx@isrl.kr, mohaisen@ucf.edu, nyang@inha.ac.kr). A preliminary version of this work has appeared in IEEE ICDCS 2017 [1].

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

this scenario, a rAP is created using the same Service Set Identifier (SSID) of a cloned victim AP to perform a man-in-the-middle attack, also known as the evil-twin. The evil-twin is not necessarily the clone of the legitimate AP providing Internet connectivity, and could be any AP in the vicinity for the attacker to clone—in this paper, and for consistency and clarity, we use "*legitimate AP*" to refer to an AP that provides Internet connectivity, and "*trusted AP*" to refer to a victim cloned by an attacker.

The recent research efforts and media reports only highlight its prevalence today, and the rAP attack has been known for many years [18]. As such, there have been various attempts to defend against it resulting in various detectors, including two notable classes of rAP detection techniques: the snooping-based [19], [20], [21], [22] and time-based approaches [3], [5], [23], [24], [25]. In the snooping-based approaches, Media Access Control (MAC) or SSID addresses of rAPs are collected in a blacklist and used later to detect them. However, this approach is fundamentally limited, since MAC and SSID addresses can be easily spoofed. On the other hand, the time-based approaches, as in [3] and [5], use the timing side-channel to detect rAPs. They assume that when packets are sent through a rAP, a relaying delay is observed, because the rAP has to use an additional wireless path for its operation. Indeed, in validating such assumption, the prior work used software-based rAPs and demonstrated a significant delay when using an AP. It has been widely accepted without validation that the observed delay is the result of the additional wireless path.

In this paper, we illustrate a limitation of the time-based techniques by showing that the delay used for inferring whether a rAP exists between a user and a legitimate AP is not due to an additional wireless path, but the result of a computational delay caused by the software bridging. We demonstrate that an adversary can manipulate this delay feature and evade detection by adopting a high-performance hardware-based *layer-2* wireless bridge with minimal bridging delay. We devise a new detection technique and demonstrate its effectiveness in detecting the proposed hardware-based rAP under the assumption that a rogue AP should use two different channels (one for relaying a legitimate AP and the other for serving stations), The assumption is verified in §3. Our detector uses two wireless interfaces: one sends a steady flow of traffic via the target AP to a remote server, while the other intentionally interferes with other channels one by one. When the target AP is legitimate, traffic obstruction caused from the interference is not observed. If our detector connects to a rogue AP using two different channels, we can observe traffic obstruction to rAP even when the interfering device is working at the other channels.

Contributions. The contributions of this work are multifold. (1) We developed a powerful rogue AP (PrAP) that defeats the existing time-based detectors and show their fundamental shortcomings. Different from the existing rAP designs using a laptop and a wireless adapter, PrAP consists of two physically interconnected off-the-shelf WLAN routers. We implemented a time-based detector [3], tested it, and showed how it fails to detect PrAP. (2) We designed PrAP-Hunter, a new detector based on a new detection assumption, namely the channel interference. PrAP-Hunter is a tool for network administrators to determine whether a given and currently connected AP is a rAP or not with high accuracy. Through extensive experiments, we show that the wireless channel communication can be interfered by intentional channel interference signals, and we quantify the throughput degradation according to the amount of channel interference. We also show that the intentional interference can be used not only to perform attacks but also to counter-intuitively and effectively defend attack from a rogue AP (e.g., rogue AP detection). (3) To the best of our knowledge, our work is the first scheme that considers channel configuration issues of the wireless bridged rogue AP. Our system detects rogue APs that are set up in a wide range of channel settings unlike previous works that can only detect attackers serving in a channel far from that of the legitimate AP. (4) We implemented PrAP-Hunter in two different hardware setups: a desktop computer and a mobile phone. We performed an extensive evaluation under various traffic scenarios using the desktop detector, and under different locations (positions) using the mobile detector. We found that PrAP-Hunter achieved close to 100% detection rate with the desktop setup, regardless of the traffic scenario. With the mobile setup, the detection rate was 100% when PrAP-Hunter was located close to the PrAP. Performance-wise, PrAP-Hunter provides a significant improvement over the state-of-the-art: Han *et al.*'s achieved 60% detection rate under heavy traffic scenarios with a software rAP, whereas PrAP-Hunter's rate is significantly higher even under a worse scenario [3]. We supplement our work with a field study for detection at 10 coffee shops. Using the mobile version of PrAP-Hunter, we executed a successful detection in all cases.

PrAP-Hunter has several advantages. (i) It can detect a hardware-based rAP that cannot be detected using time-based rAP detectors. (ii) It works without requiring any prior knowledge of information such as the SSID, MAC address, Received Signal Strength Indicator (RSSI), and clock skew on the examined network. (iii) It provides significantly higher detection rates even under a heavy traffic scenario. (iv) It is fast, and the detection of a rAP is completed within 10 seconds. (v) It is cheap; implemented on a smartphone with an additional off-the-shelf WLAN card.

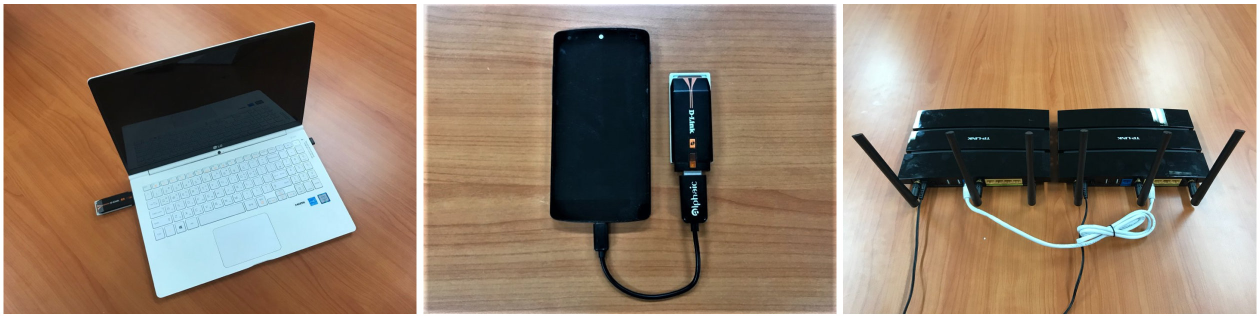
Organization. The organization of the rest of this paper is as follows. The related work is described in §2. The threat model is outlined in §3. The detection strategy is outlined in §4. Our detector for rogue APs is described in §5. Our experimental setup is described in §6. In §7, we present our experimental results and performance evaluation. Real-world deployment and testing are addressed in §8, while concluding remarks are drawn in §10.

2 Related Work

Rogue AP (rAP) detection methods are mainly classified into two categories: snooping-based [19], [20], [21], [22], [26], [27], [28], [29], [30] and time-based detection [3], [5], [23], [24], [25]. The snooping-based schemes use sensors to collect features of APs, e.g., SSID, MAC address, channel, RSSI, and clock skew. The collected features are then compared with previously known features of rogue (or legitimate) APs to determine the legitimacy of a given AP. The second category of schemes depends on the characteristics of inter-packets, the round trip time or traffic to detect rAPs. Generally, those techniques do not require any prior knowledge about the wireless devices, but sometimes they need to configure site-specific parameters for better detection rate. These schemes can actively detect a rAP by collecting the required information in real time.

Snooping-based approaches. In the snooping-based approaches, a prior knowledge is used to detect the presence of a rAP. In [19], [21], [22], the MAC address of an AP is compared against addresses of known APs for detection. An unknown MAC address indicates that an AP is rogue. Also, other nonforgeable factors like RSSI values [27], [29], clock skew [28], [30], or radio frequency

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) Laptop/USB (b) Phone/USB (c) WAN/WLAN (PrAP)

Figure 2. Three ways ways of setting up a rogue AP (rAP), using a universal serial bus (USB)-based wireless interface (along with the built-in wireless interface) as in (a) with a laptop and (b) with a phone, and using a physical layer connector of two routers (PrAP) as in (c).

variations [26] are used to fingerprint rAPs. While easy to use as detection features, it is well known that features such as identifiers, including the SSID and MAC address, can easily be spoofed. Further, those approaches are costly in requiring equipment setup for collecting such a prior knowledge (*i.e.,* authorized list), or collecting data (*e.g.,* RSSI, RF wave) from traffic sensors.

Time-based approaches. Beyah *et al.* suggested a method that utilizes temporal characteristics, such as inter-packet arrival time [24]. Wei *et al.* [31], [32] proposed two similar detection schemes by examining the arrival time of consecutive ACK pairs in TCP traffic. Watkins *et al.* and Qu *et al.* [33] used the round trip time of TCP traffic, Venkataraman *et al.* [34] based their approach on DCF pattern in the wired traffic, while Mano *et al.* [35] based their approach on physical properties of half duplex channels for detection. All the above works are for detection of wired rogue APs, but for wireless rogue APs, there are two known works: one is by Yang *et al.* [5], and the other by Han *et al.* [3]. Yang *et al.* proposed an "evil twin" detector using a discriminative feature of inter-packet arrival time of a rAP [5]. Han *et al.* developed a time-based detection technique that uses RTTs through additional wireless line [3]. These two techniques use packet delay of traffic caused by the rAP as a feature for detection. While they have a lower cost than snooping-based approaches, since they do not require any setup of any additional sensors, these schemes are sensitive to network conditions, with network instability causing spikes in the false alarm rates.

Other approaches. Lanze *et al.* [7] fingerprinted software rAPs using attack tools properties (e.g., MadWifi [36], aircrack-ng [37], Karma [38]). Kindberg *et al.* [39] and Roth *et al.* [40] proposed two authentication-based approaches that additionally require the user interaction. In [39], users required to check key management results shown on an additional display belonging to the legitimate AP. Similarly, in [40], the established key was used to encode a short string as a sequence of colors, and rendered in both the user device and the legitimate AP. While efficient, these works do not support multiple users authenticate at the same time. Bauer *et al.* [41] suggested to mitigate evil twin attacks by using the contextual information which is defined by the information of nearby APs. This work was based on Trust on First Use (TOFU), which has potential vulnerabilities when associating with an AP for the first time. Later, Gonzales *et al.* [42] improved Bauer *et al.*'s work by combining the RSSI with the context. Moreover, they adapted Pang *et al.*'s "wifi-report" system [43] to reduce the risk of TOFU. Gonzales *et al.* also suggested an SSH-like authentication method to secure data delivery. However, modifications of EAP protocol were required. Our threat model is more realistic by not requiring TOFU or infrastructure modification. To this end, the literature focused on software-based rAPs, and proposed detections for them. In this paper, we introduce a powerful hardware-based rogue access point (PrAP) that can evade time-based rAP detectors, by avoiding the timing channel, and evade the snooping-based methods by spoofing their detection information (e.g., MAC and SSID), and by turning off the broadcast of beacon frames. To address this PrAP, we propose a channel interference-based PrAP-Hunter. Channel interference is a kind of jamming, which has been treated to be defeated as in [44], but we use the jamming in a positive way to detection.

3 Threat model

A network administrator needs to check whether an AP in the enterprise network is a trusted AP or a rogue AP (rAP). Regular check-up of rAPs are desirable because users carry out confidential communication over an AP that they believe trustful. In this paper, a rAP is defined as an AP that relays WLAN traffic between *a legitimate AP* providing Internet connectivity and a station, and may act as a man-in-the-middle trusted AP of which device information is cloned from *a trusted AP*. To this end, we assume that a rAP has two wireless interfaces, one connected to the legitimate AP in station mode and another disguised as the trusted AP in service mode. When a user connects to the rAP, two interfaces will forward traffic back and forth. This relaying attack has been reported in [17].

Software-based rAP. In the literature, rAPs are defined using a laptop and an additional WLAN USB adapter, as shown in Figure 2(a) [3], [4], [5], [7]. This type of rAP can easily be set up by adding rules to the `iptable` or by setting up Internet sharing functionality of Microsoft Windows or Mac OS. As shown in Figure 2(b), with the development of smart devices, we can setup a rAP by utilizing a mobile phone and an additional WLAN USB adapter, since some customized ROMs support WLAN connection with on-to-go (OTG) cable. Configuring a rAP with a laptop or a mobile phone can give an adversary portability features. However, such rAPs relay packets between two wireless interfaces in a software-based approach. Therefore, the performance of such APs depends on the computational power of the software bridging. We proved in §7 that the time-based approach proposed by Han *et al.*'s [3] can detect software-based rAPs only, but not the powerful hardware-based rAP (PrAP) having little bridging delay. Also, Lanze *et al.* [7] outlined an effective method to fingerprint such rAPs.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Hardware-based PrAP. Figure 2(c) shows a setup of a PrAP costing under $100, and achieving high performance in relaying packets between two wireless interfaces in a hardware-based approach. The PrAP is characterized by a low delay, and is difficult to detect using time-based rAP detection methods. Moreover, the plentiful capacity (*i.e.,* 1 Gbps) of mirroring port of PrAP helps capturing raw packets and injecting manipulated packets without packet loss. We note that the mirroring port does not delay the packet relaying pipeline (See details in Section 6.1).

3.1 Assumptions

An attacker in our threat model is assumed to wirelessly connect to a legitimate AP (*i.e.,* Internet provider) and to clone all information of one of the trusted APs (*i.e.,* cloned target) except the channel information. The Internet provider can be one of the APs around the attacker, including the cloned target APs in the enterprise network or any other APs providing Internet connection. We consider various cloning scenarios to explain our assumption of the channel used by adversaries.

As a way of replacing the two interfaces with one interface, one may assume as well the ability of the adversary to plug a rAP directly in the backbone, thus eliminate the need for using a second channel, which is utilized in our approach for detection. First, we point out the large body of the literature [23], [24], [25], [31], [32], [33], [34], [35] that address the problem directly with wired rAPs. Moreover, we also point out that the assumption made is very strong and often impractical. A plugged rAP in most enterprises would be visible, and can be detected through physical security measures. Even where that is possible (*e.g.,* an employee unplugging his PC with a dedicated IP, and replacing it with a rogue access point), plausible scenarios leading to this kind of attack are out of the scope of our threat model: they would require an insider attacker, which is way beyond the attack capabilities we assume in our work.

3.1.1 Basic assumption

Our basic assumption is that adversaries clone the SSID, MAC address and password of a target AP. There are several reasons for assuming and justifying how adversaries can clone the password of the target APs. First, this assumption is necessary for the operation of the rogue AP. For example, if the rogue AP is to use a different password than the one known to users using a public (legitimate) AP, connections by victims will be automatically rejected. While one can cope with this issue (*e.g.,* making the AP public or programmatically modifying the AP to accept any password), not needing to copy the password, such a mitigation would require modifying the AP. Moreover, when getting rid of the password altogether and making the AP public, recent operating systems (*e.g.,* Microsoft Windows, Apple's iOS and Google's Android) alert users about unsafe wireless connections when accessing public APs without authenticating. Second, the justification of being able to clone the password is quite straightforward in today's wireless access points usage. For example, in public spaces and facilities, such as restaurant, hospital, shops, public transportation, lounges, *etc.*), the WiFi password is often posted on the wall of the facility. As such, it is easy for an adversary to obtain the password of the legitimate AP. Finally, as mentioned in reasoning about the assumption, a determined adversary can build his own RADIUS server that approves all accesses without verifying the credentials, which has the equivalent effect of password cloning.

3.1.2 Cloning channel of Internet provider APs

An adversary should avoid cloning the channel of an Internet provider AP because of the co-channel interference problem. In our work, the rogue AP consists of two wireless interfaces. According to Villegas *et al.* [45], a channel sharing by two wireless interfaces degrades a 6 Mbps traffics by around 50% (*i.e.,* 3 Mbps), even when they are placed in different rooms (*i.e.,* when having a large spatial distance), and under light network conditions. Moreover, Zubow *et al.* [46] proved that the distance between the wireless interfaces is also an important factor affecting the interference. Per to their results, a smaller spatial distance (*e.g.,* less than 1 meter) between two wireless interfaces leads to a much stronger channel interference. Because the spatial distance for a rogue AP is usually limited (*e.g.,* they two interfaces have to be contained in proximity), the rogue AP must increase the channel interval between wireless interfaces for avoiding the self-interference.

3.1.3 Cloning channel of target APs

To observe what happens when a rogue AP clones all information of the target AP (*i.e.,* SSID, MAC address, channel, the security protocol (WPA, WEP), encryption (AES, TKIP), and the password), we conducted an experiment, where two smartphones running different operating systems (*i.e.,* Android and iOS) established wireless connections with those identical access points. We found that only one SSID could be probed by both smartphones, which means the PrAP could successfully disguise itself by cloning the information of the legitimate one. However, both smartphones eventually failed to establish a connection with either of the APs. By examining the wireless management frames (Layer 2), we found that the client failed in the four-way handshake step of establishing a key with the access point. The connection failure happens because all key exchange frames from the client are heard by both the rogue and the legitimate APs. In establishing a session key, we note that both the rogue and the legitimate APs reply back to the smartphone frames with different random keys (*e.g.,* K1 and K2). Then, the client replies back with the first frames (*e.g.,* K1) it received, leading the AP that sent the frame with K2 and received the wrong reply for K1 to send back a connection failure message to the station. In conclusion, the station cannot make the connection to the rogue AP on the same channel.

3.1.4 Channel use of PrAP (Stronger Adversary)

It is also a common practice when providing wireless services in a large area (*e.g.,* enterprise, campus, shop, *etc.*) for network administrators to set up several APs with the same SSID operating on different channels. Having multiple APs and running them on different channels (*e.g.,* 1, 6, 11) is important to cover a whole area with a strong RSS, to address separate workloads while avoiding channel interference. Moreover, multiple APs within close proximity (*e.g.,* three legitimate APs serving on channels 1, 6 and 11, and using the same SSID but different MAC addresses) can be forced to use the same channel and even the same MAC. However, as shown earlier, any connection cannot be established in this scenario as the four-way handshake procedure in the session key establishment would fail, which also would be the case for our rAP if operated on the same channel as one of the legitimate APs. As such, and in order to address this issue in a setting with multiple legitimate APs, the adversary would clone the information of the legitimate AP on channel 1, and serve on channel 6 or 11 by relaying through channel 1. As a result, rAP would still provide internet service without self-interference.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

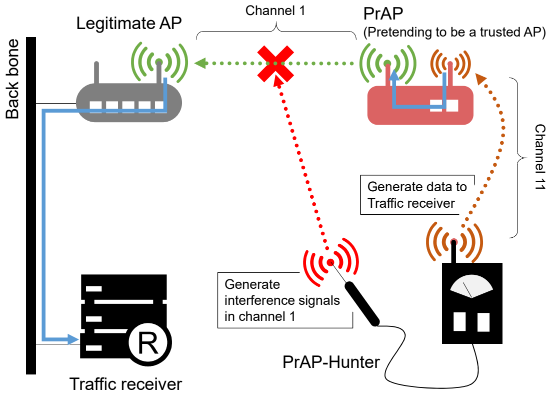

Legitimate AP
Channel 1
PrAP
(Pretending to be a trusted AP)
Back bone
Channel 11
Generate data to Traffic receiver
Generate interference signals in channel 1
PrAP-Hunter
R
Traffic receiver

Figure 3. A legitimate AP on channel 1 and a PrAP repeating signals of the legitimate AP on channel 11. The PrAP-Hunter generates traffic to the traffic receiver through the PrAP. The interference device interferes with channel 1. An AP is said to be rogue if we observe obstruction of traffic on channel 11 via the PrAP-Hunter.

Table 1
Bandwidth overlap against channel gap

Channel gap	Overlapping bandwidth
0 (ch1 vs. ch1)	20 MHz
1 (ch1 vs. ch2)	17 MHz
2 (ch1 vs. ch3)	12 MHz
3 (ch1 vs. ch4)	7 MHz
4 (ch1 vs. ch5)	2 MHz
5 (ch1 vs. ch6)	0 MHz

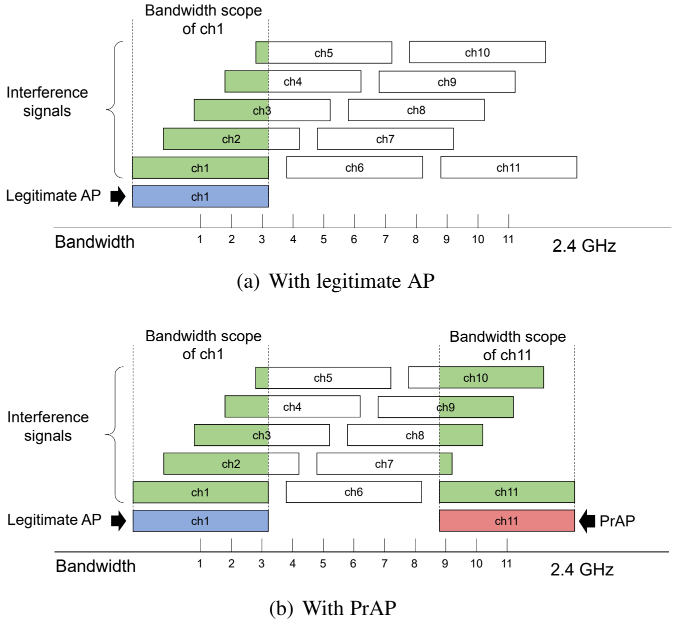
4 Detection Strategy

4.1 The Basic Concept

Our PrAP-Hunter has two wireless interfaces, one that associates itself with a target AP to generate traffic to a receiver (a server listening TCP connection) during detection, while the second interface (interference device) interferes with channel 1 to 11 sequentially with a rest time. Figure 3 illustrates how the proposed method works. The PrAP-Hunter connects to the target AP (ch 11), which relays signals between the legitimate AP (ch 1) and a PrAP-Hunter (ch 11). In the beginning PrAP-Hunter does not know whether the AP is relaying signals or not. When the PrAP-Hunter generates traffic to the receiver, both channels 1 and 11 contribute to the data transmission. From the standpoint of PrAP-Hunter, if obstruction of data transmission is observed at channel 11 when the interference device interferes with channel 1, this is a strong indicator that the target AP is relaying signals wirelessly. When that happens, the connected AP must be a PrAP.

4.2 Channel Interference in 802.11n

As described in the 802.11n standard, the channels used for WLAN are separated by 5MHz in most cases, but have a bandwidth of 20MHz. In other words, each channel shares bandwidth with other adjacent channels. Considering a 20MHz bandwidth channel, there is 17MHz of bandwidth shared between channels 1 and 2, and 2MHz of bandwidth shared between channel 1 and 5 (Table 1). It means when the interference device works on a certain channel, it does not only interfere co-channel, but it also interferes the adjacent channels sharing the bandwidth.



(a) With legitimate AP

(b) With PrAP

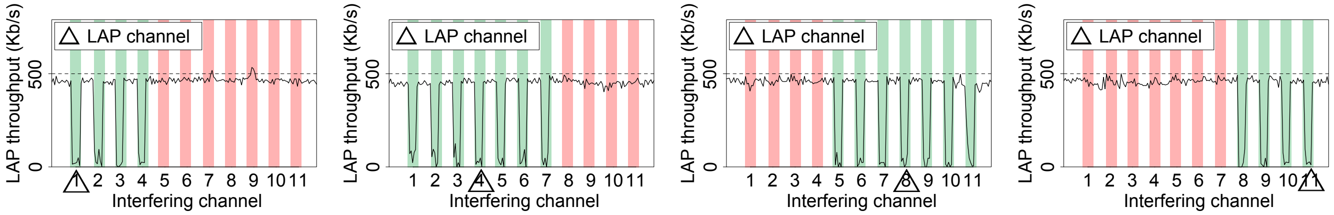
Figure 4. Channel interference under IEEE 802.11n

4.3 Advanced Detection Strategy

Figure 4 shows our PrAP detection strategy, considering the wireless bandwidth standpoint. In Figure 4(a), we show a detection scenario where the legitimate AP uses channel 1 and no PrAP exists. We generate traffic through the currently connected AP, while the interference device is transmitting data on channel 1 to 11 with a rest time between each channel interference. When the interference device transmits on channels 1 to 4, the throughput of the legitimate AP at channel 1 is obstructed because of bandwidth sharing as shown in Figure 5(a). Channel 5 also shares a 2MHz bandwidth with channel 1, but 2MHz bandwidth sharing is not enough to interfere substantially. Also, based on both the work in [46] and our experiments, if the PrAP-Hunter and the interference device are located farther than 50cm apart, channel interference caused by 2MHz bandwidth sharing is insignificant. As a result, we obtained Figure 5(a). Throughput degradation for the other channels by the interference are shown in Figure 5, and the channel overlapping shown in Table 1 is confirmed. That is, a channel *ch* is interfered by data transmission over channels from *ch*-3 to *ch*+3 (6 in total, *ch* excluded). For example, throughput on channel 5 would be obstructed by transmission over channels 2,3,4 and 6,7,8. Figure 4(b) shows a detection scenario where a PrAP (ch 11) repeats a signal of legitimate AP (ch 1). If the AP being connected on channel 11 was a legitimate AP, the results of detection should look similar to the results reported in Figure 5(d). However, because we experienced an unexpected throughput degradation on channel 11 as shown in Figure 6(d) when we interfered over channels 1-4 (throughput degradation should have occurred only when interfering over channels 8-11 without a PrAP), we conclude that the connected AP is a PrAP, and it provides wireless connectivity by repeating signals.

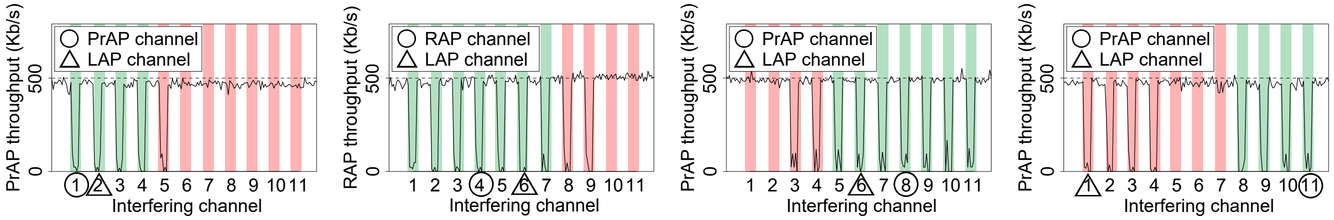
To avoid being detected by the obvious throughput degradation by data transmission over unassociated channels, attackers may set a PrAP in an adjacent channel of a legitimate AP, as in Figure 6(a). In this scenario, a PrAP set up at an adjacent channel (ch 1) to a legitimate AP's channel (ch 2). Even in this case, we could observe that transmission over channel 5 also obstructed

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) Channel 1 (b) Channel 4 (c) Channel 8 (d) Channel 11

Figure 5. Cases of only a legitimate AP on various channels. Green bars indicate the overlapped channels with the connecting AP's (here, a legitimate AP) channel affected by interferance, which confirms the channel overlapping model of IEEE802.11n.



(a) (PrAP, legitimate AP) → (1, 2) (b) (PrAP, legitimate AP) → (4, 6) (c) (PrAP, legitimate AP) → (8, 6) (d) (PrAP, legitimate AP) → (11, 1)

Figure 6. Cases of a PrAP existence varying the channel of the PrAP and of the legitimate AP. Red bars indicate the non-overlapped channels. The non-overlapped channels are affected by interference with the channel of the connecting AP (here, a PrAP).

the connected AP's channel (ch 1). The interfering channels, thus, were from 1 to 5 (Figure 6(a)), which were different from 1 to 4 in the legitimate AP case (Figure 5(a)). Therefore, using the interference information, we can infer that the connected AP used not only channel 1 but also channel 2. Simply put, if the number of obstructed channels is more than that of the legitimate AP's only scenario (that is, the number of throughput degradation in Figure 6 is greater than that in Figure 5), there must be a PrAP.

4.4 Implementation Issues

Data transmission. When traffic is generated between the PrAP-Hunter and the traffic receiver, we should guarantee the stability and the speed of the traffic to improve detection accuracy. The location of the traffic receiver is the most important factor in transmission stability, which is affected by traffic volume and dynamic routing paths. Increasing the transmission rate can improve the success rate of detection, but it also increases the workload of the AP, which may affect communication with other users.

Duration and level of interference. If interference with a target channel is too long, it causes nearby devices operating in adjacent channels to experience network delays. This should be avoided by adjusting both the duration and level of interference.

Type of interference message. We need a message that requires a large amount of data, and can guarantee stable interference signals. Also, messages should be broadcast to all devices, because which legitimate AP is used by the PrAP is not known in advance.

Distance between devices. Channel interference is caused by the overlap of adjacent channels [46]. Also, the distance of TX-TX or TX-RX devices affects the interference range of adjacent channels.

5 PRAP-HUNTER

We outline a method to derive the degree of channel interference (§5.1), steps to obtain parameters for effective interference (§5.2), and our detection algorithm (§5.3).

5.1 Channel Interference Degree

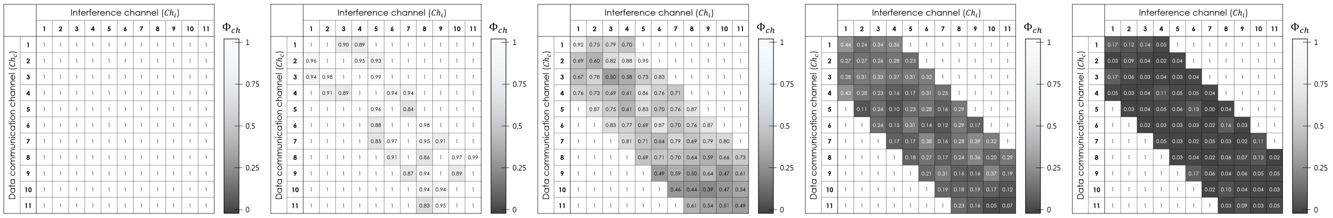
We show how to derive the degree of channel interference, Φ, which is necessary for the operation of PrAP-Hunter. During detection, PrAP-Hunter generates stable traffic to the receiver through the currently connected AP channel (ch_{ap}) and records changes in throughput over regular time intervals. Simultaneously, an interference device generates noise through interference channels (ch) 1 to 11 sequentially. Φ_{ch} is a throughput index of channel ch_{ap} for an interfering channel ch. It has a lower value when transmission over ch does not interfere with the channel ch_{ap}, but a higher value when transmission over ch interfere effectively.

Before each channel interference, the PrAP-Hunter has some rest time for traffic recovery. The PrAP-Hunter calculates the mean throughput during the rest time as ntm_{ch} (normal throughput mean). The PrAP-Hunter also calculates the mean throughput of the AP during the channel interference with ch_{ap} via ch as itm_{ch} (interference throughput mean). Using itm_{ch} and ntm_{ch}, we define the degree of channel interference Φ as

$$\Phi_{ch} = \frac{\mathsf{itm}_{ch}}{\mathsf{ntm}_{ch}}. \qquad (1)$$

In this paper, we use a fixed threshold value of 0.5 for Φ_{ch} to determine whether data transmission through the currently connected AP channel (ch_{ap}) is being interfered by noise through the interference channel (ch). From the observation in our experiments, it is hard for Φs to reach values under 0.5 without inducing an intentional channel interference (even when we sent 144 Mbps traffics through the legitimate AP), because the generated traffic volume for detection is quite small. Our system works well in various traffic congestion situations with our threshold as will be shown in §7. If Φ_{ch} is less than 0.5, we determine that the data transmission is being interfered with by noise through ch. Otherwise, we determine that the data transmission is not being interfered with by noise. After interfering with all channels, we obtain Φ values for all of the 11 channels; namely we obtain $\mathsf{R} = \{\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_{10}, \Phi_{11}\}$.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) 30Mbps at 30FPS. (b) 30Mbps at 40FPS. (c) 30Mbps at 50FPS. (d) 30Mbps at 60FPS. (e) 500Kbps at 250FPS.

Figure 7. Examining the performance of modified beacon frame's interference under different conditions. Shaded blocks are affected channels by interference. Darker blocks represent more interference.

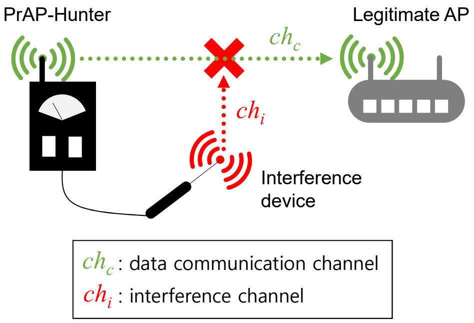

PrAP-Hunter
Legitimate AP
ch_c
ch_i
Interference device
ch_c : data communication channel
ch_i : interference channel

Figure 8. Traffic sent via data communication while beacon is sent via interference channels.

Depending on the being used AP channel (ch_{ap}), we separate a set of interference degrees from R such that

$$\mathsf{B} = \{\Phi_{ch} || \mathsf{ch}_{ap} - ch| > 3, \Phi_{ch} \in \mathsf{R}\} . \quad (2)$$

B includes Φ_{ch}'s, where interference channel (ch) of Φ_{ch} has a channel gap of more than 3 compared with ch_{ap}. That is, we consider only channels (ch) that have no bandwidth shared with ch_{ap}. Why the number "3" is was explained in section 4.3 using Table 1. The reason we care only the channels that is far from ch_{ap} by more than 3 channels is to check whether the throughput on ch_{ap} is obstructed or not by noise transmission over the independent channels from ch_{ap}. If the AP is a legitimate AP, we cannot obtain a Φ_{ch} in the set B that is less than our threshold of 0.5. However, if it is a PrAP, we will obtain at least one Φ_{ch} in the set B that is less than the threshold of 0.5. For evaluating the performance of our method, we collect the minimum Φ_{ch} in the set B after each detection, and denote it by Φ_{min}.

5.2 Efficiency and Impact of Interference

For effective intentional channel interference, we use a modified beacon frame, which will be described in §6.4. The basic function of the beacon frame is to broadcast signals of existence and connection information of the AP to stations. However, beacon frames broadcast too frequently by APs also can be problematic, and may increase the workload of the surrounding devices.

The degree of channel interference depends on the frame rate of the interfering beacon and the data transmission rate of the traffic generator. Figure 7 shows the results of channel interference experiments under various conditions of the experimental setup in Figure 8. In this figure, the number in each block denotes the degree of channel interference, and shaded blocks are the channels affected by interference. Darker blocks represent more interference. As shown in Figure 7(a), at data transmission rate of 30 Mbps over the connected AP channel (Ch_c) and interfering beacon of 30 FPS over the other interfering channels (Ch_i), we did not observe any transmission obstruction, even when interference was sent through channels that had bandwidth shared with the data transmission channels. As shown in Figure 7(b), at 30 Mbps and 40 FPS we observed transmission obstruction with interference signals through a bandwidth-sharing channel. However, degrees of channel interference were not clear enough to conclude that data transmission was obstructed by the interfering beacon frames, because we obtained similar results in some unstable networks. As shown in Figure 7(c), at 30 Mbps and 50 FPS, transmission is obstructed when interference was sent through channels that have bandwidth shared with the data channels. Similar to results in Figure 7(b), we also observed unstable channel interference. Figure 7(a)-(d) show that when interference was sent through channels that had bandwidth sharing, the degree of channel interference was affected by the frame rate of the interference. Figure 7(d) shows that with interference of 60 FPS (frame per second) we could stably interfere with data transmission when an interference signal was sent through bandwidth sharing channels.

It seemed at first to be more advantageous to use a higher transmission rate over Ch_c. However, we noticed that high transmission rate might increase the workload at an AP, and might affect the experience of other users negatively precluding it from real deployment scenarios. Also, we noticed that the data transmission rate significantly depends on the network state and the performance of the AP; the data transmission rate cannot be guaranteed at a high rate. For these reasons, it would be better if we are able to interfere effectively with a lower data transmission rate. Upon various attempts of adjusting the parameters, we obtained the experiment results shown in Figure 7(e), where an interference of 250 FPS effectively worked, even at low data transmission rates (500 Kbps). High-speed beacon transmission also affected other devices that listened to the beacon and increased the error rate of data transmissions. Thus, we need to minimize channel interference time to avoid such side-effects.

5.3 Detection Method

The PrAP detection consists of three algorithms: a PrAP-Hunter, an interference algorithm, and a traffic receiver. We run Algorithm 1 to determine whether the used AP is a PrAP or not.

PrAP-Hunter. Algorithm 1 presents the PrAP-Hunter, consisting of preparation, interference repeating, and traffic analysis phases. The first phase implements the preparation for detection. Connect first makes association to the target AP where the SSID is SSID_{ap} and obtains channel information ch_{ap}. After association, Send_B, a blocking IO function, builds a TCP/IP connection via ch_{ap} with the traffic receiver (TrafficReceiver) and sends random data ($data$) for Δt time. This is done to check the state of the TCP/IP connection and ensure data transmission rate stability for the

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

Algorithm 1 PrAP-Hunter

```
Input: SSID$_{ap}$
Output: true/false
/* traffic measurement */
ch$_{ap}$ ← Connect(SSID$_{ap}$)
$thput[0]$ ← Send$_B$(ch$_{ap}$, TrafficReceiver, $data$, $\Delta t$)
for $ch$ ← 1 to 11 do
    $thput[ch]$ ← Send$_{NB}$(ch$_{ap}$, TrafficReceiver, $data$, $\Delta t$)
    Interfere$_B$(InterferenceDevice, $0.6\Delta t$, $0.4\Delta t$, $ch$)
end for
/* traffic analysis */
for $ch$ ← 1 to 11 do
    if |ch$_{ap}$ − $ch$| > 3 then
        /* normal throughput */
        ntm$_{ch}$ ← Mean($thput[ch][0 \sim 0.6\Delta t]$)
        /* throughput under interference */
        itm$_{ch}$ ← Mean($thput[ch][0.6\Delta t \sim \Delta t]$)
        $\Phi_{ch}$ = itm$_{ch}$/ntm$_{ch}$
        if $\Phi_{ch} < 0.5$ then
            return false        /* PrAP */
        end if
    end if
end for
return true        /* Legitimate AP */
```

Algorithm 2 Interfere$_B$

```
Input: $ch$
Wait($0.6\Delta t$)
SetChannel($ch$)
$t_1$ ← CurrentTime()
while $(t_2 - t_1) \leq 0.4\Delta t$ do
    Broadcast($ch$, $modifiedbeacon$)
    $t_2$ ← CurrentTime()
end while
```

next phase. In the second phase, the PrAP-Hunter repeats the channel interference. For each round of interference, Send$_{NB}$, a non-blocking IO function, sends $data$ to TrafficReceiver for Δt time and records the throughput in $thput[ch]$ during that period of time. The PrAP-Hunter then executes a blocking IO function, the InterferenceDevice rests for the first $0.6\Delta t$. After that, Interfere$_B$ (Algorithm 2) is executed to interfere with ch for $0.4\Delta t$. Thus, the data structure $thput[ch]$ has the throughput recordings for two parts: the first $0.6\Delta t$ is where there is no interference, and the subsequent $0.4\Delta t$ is for recordings under interference as a result of applying InterferenceDevice on ch.

The last phase is for traffic analysis. The detector calculates the mean normal throughput ntm$_{ch}$ for the period with no interference, and the mean throughput itm$_{ch}$ for the period with interference, and calculates the degree of channel interference as in (1). If Φ_{ch} is less than the threshold of 0.5, we conclude that data transmission via ch$_{ap}$ is obstructed by interference of the other channels, so the AP connected is a PrAP. When any data obstruction at ch$_{ap}$ is observed for the entire period of interference, we conclude that the AP is not a PrAP.

Interference. The algorithm 2 presents the interference procedure. In our method, the interference device does not interfere with a specific AP, but with a specific channel signal, or with all APs using that channel. Therefore, broadcasting frames such as beacon (as opposed to destination-designated frames) fit our purpose. When the PrAP-Hunter starts, the interference device is put in a standby mode waiting for command from the PrAP-Hunter. When the interference device receives channel information ch, it is put into the standby mode for $0.6\Delta t$. After the $0.6\Delta t$ time has passed, it starts broadcasting modified beacon frames for $0.4\Delta t$.

Receiver. The third algorithm, TrafficReceiver, receives data generated by the PrAP-Hunter. TrafficReceiver waits for connections from the PrAP-Hunter. When it receives data from PrAP-Hunter, the receiver discards it to avoid unnecessary waste of resources.

6 Experimental Setup

We implemented PrAP-Hunter in two settings: a high-end hardware (desktop) in a fixed position for analyzing the performance under various traffic scenarios and a mobile PrAP-Hunter was implemented on a relatively low-performance mobile device and is used for analyzing the performance under various locations.

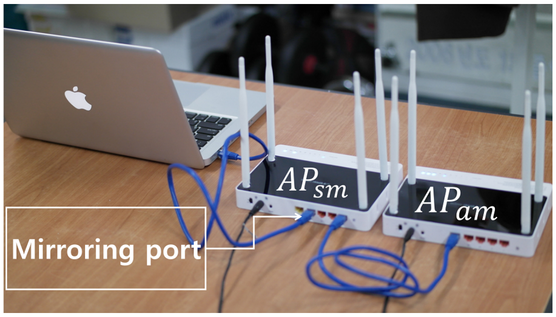


Figure 9. PrAP's Hardware. AP$_{sm}$ is a WLAN router associated with a legitimate AP, and AP$_{am}$ is a WLAN router disguised as a legitimate AP.

6.1 Legitimate AP and PrAP

An EFM ipTIME N8004R is used in our experiments to setup the legitimate AP (in 802.11n mode). In this work, we focus on a PrAP that very quickly repeats the signals of a legitimate AP. The PrAP consists of two WLAN routers (EFM ipTIME N8004R), where one is in the station mode (AP$_{sm}$) and the other is in an AP mode (AP$_{am}$). Figure 9 shows the PrAP used in this paper. In this figure, AP$_{sm}$ is responsible for repeating signals to and from the legitimate AP. AP$_{am}$ and AP$_{sm}$ are interconnected using a LAN cable, and AP$_{am}$ is assigned a valid IP from a DHCP server of AP$_{sm}$ with a spoofed SSID and MAC address. Attackers could plug a LAN cable into a port of AP$_{am}$ or AP$_{sm}$ for a port mirroring function that helps data capture much easier. All devices are operated in the IEEE 802.11n mode with MIMO.

6.2 Desktop Detector

The hardware configuration of our desktop PrAP-Hunter is a PC with an Intel Core i5-3570K CPU, 4GB RAM, an ipTIME n500U external wireless card as a traffic generator, and a D-Link DWA-125 external wireless card as an interference device (Figure 10). We implemented our PrAP-Hunter using C# in MonoDevelop (ver.2.8.6.3) supporting a GUI development environment in Linux Ubuntu 12.04 (kernel ver.3.2.0-33-generic). The interference device was implemented in C with the Loss of Radio Connectivity (Lorcon2) library, which is a generic library for injecting 802.11 frames in the MAC layer. Lorcon2 allows modifying 802.11 frames to inject frames through specific channels. As shown in [46], the distance between devices is also an important interference factor. To maintain the same interference conditions, we placed the interference device at the same distance as the PrAP-Hunter, the legitimate AP, and the PrAP, as shown in Figure 11.

6.3 Mobile PrAP-Hunter

Figure 12 shows the hardware configuration of our mobile PrAP-Hunter, which consists of a Google Nexus 5 LG-D821 with a TP-LinkTL-WN722N external wireless card for interference. We used the internal wireless card associated with the mobile device as a traffic generator. For the software, we implemented

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

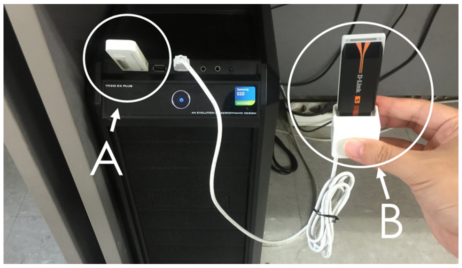


Figure 10. Hardware setting of the desktop PrAP-Hunter. A is an wireless interface that is connected to the target AP and generates traffic, and B is the wireless interface that sends interference signals through 2.4GHz channels.

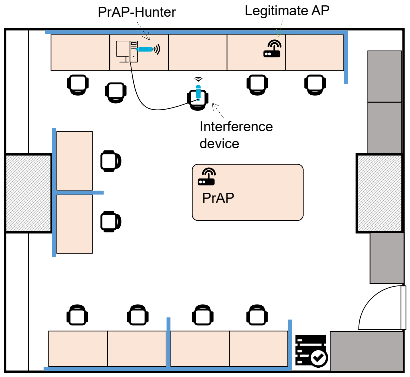


Figure 11. Map of the experiment for the desktop PrAP-Hunter.

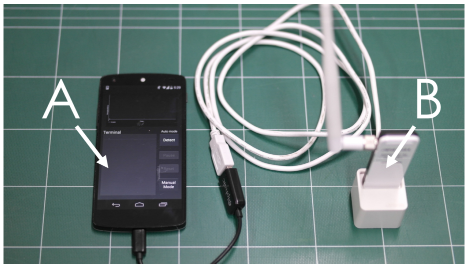


Figure 12. Hardware setting of the mobile PrAP-Hunter.

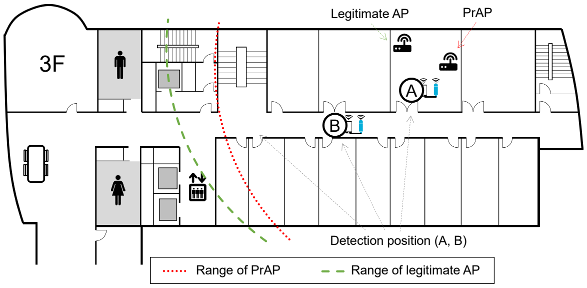


Figure 13. Map of the experiment for the mobile PrAP-Hunter.

the detector with an Android application running Omni-4.4.2-20140513-hammerhead-NIGHTLY with kernel 3.4.0-ElementalX-0.21+. The interference device was implemented in C. The PrAP-Hunter communicates with the interference device through JAVA secure channel (Jsch) library. Cross-compiled Lorcon2 and libpcap libraries were also used for running the interference device.

Experiments are performed in different positions: A and B, as in Figure 13. In position A, the distance between each device was identical to the desktop experiment. With the experiment in position A, we tried to verify the accuracy of the mobile PrAP-Hunter. We chose position B to perform our experiments and to analyze if the position of the PrAP-Hunter affects the results.

6.4 Modified Beacon Frame

In reality, most AP devices construct beacon frames less than 500 bytes in size. However, we needed a beacon frame that contained large amounts of data to stably generate interference signals. Thus, we modified the size of the beacon frame to contain up to 1500 bytes. For sizing up our beacon frame, random information is added in the network data field.

6.5 Time of Detection

A timer was used to record traffic and the interval was set to 0.2 seconds (s). In our experiments, we set Δt to 5s (3s for traffic recovery and 2s for interfering). Furthermore, we interfered with all channels. However, considering that interfering with a channel ch also affects the adjacent *six* channels (from $ch-3$ to $ch+3$) owing to the channel overlapping property as shown in Table 1, we do not need to interfere with all channels but with only 2 channels. Thus, we spend 5s at a minimum and 10s at a maximum. When using 5GHz bandwidth, channels do not share bandwidth between each other. Thus, the detection time is the number of channels in the 5GHz multiplied by Δt. Moreover, in the mixed case (i.e., 2.4 GHz and 5.0 GHz), the total detection time is the sum of 2.4 GHz and 5.0 GHz detection times.

7 Evaluation

7.1 Evaluation of PrAP

To evaluate the performance of the PrAP in context, we implemented a time-based detector described by Han *et al.* [3], where they used the round trip times between station and a DNS server and between station and AP to determine whether the used AP is rogue or not. They stated an additional wireless interval led to delay in the round trip time of a DNS query. Their rAP was software-based. We argue that the observed delay was not the result of an additional wireless path, but rather the result of a computational delay caused by the software bridging.

To show that, we performed experiments for Han *et al.*'s algorithm under the rAP and the PrAP. the rAP was configured as in Figure 2(a) and as described in [3] (a software-based rogue AP), and the PrAP was configured as shown in Figure 2(c) (a hardware-based rogue AP), which is the one developed in this work. Figure 14 shows that Han *et al.*'s algorithm could successfully distinguish the legitimate AP and the software-based rAP. However, we also see that the same technique did not work against the hardware-based PrAP (i.e., the mean of Δt is mixed for both the legitimate AP (blue circles) and the PrAP (red crosses). Yang *et al.*'s work also tried to solve the same problem using inter-packet arrival time (IAT) [5]. Although they distinguished one-hop IAT from two-hop IAT to detect the relay attacker, the empirical values to distinguish them were much greater than the theoretical values, which again implies that there is hidden delay caused by the software-based relaying.

7.2 Desktop PrAP-Hunter

Figure 15 summarizes the results of our experiments in an idle and a heavy traffic scenarios. For the idle traffic scenario, experiments were conducted around 3:00 AM at an office space. For the heavy traffic scenario, we used two wireless adapters to generate maximal data rate of 144 Mbps through the legitimate AP, which is

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

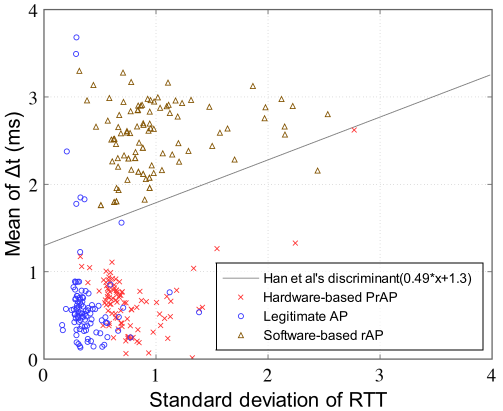


Figure 14. Results of Han *et al.*'s [3] algorithm for two rogue APs, a software-based rAP and a hardware-based PrAP.

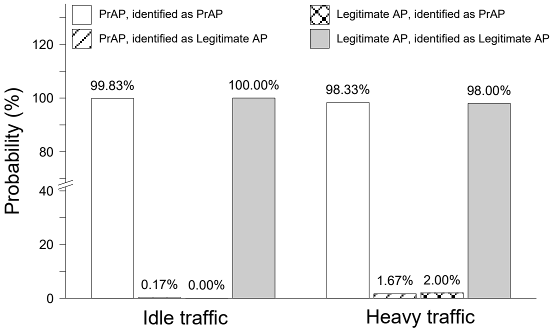
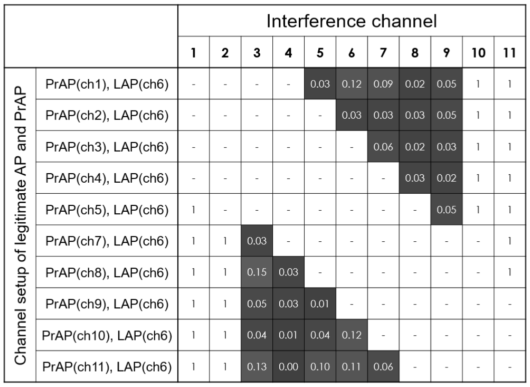
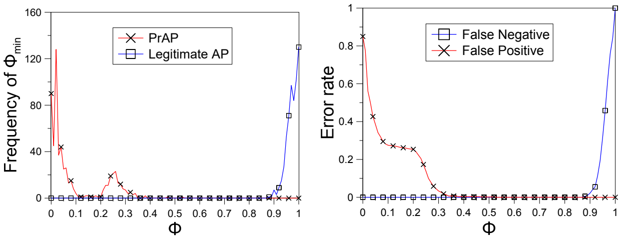


Figure 15. Examining the accuracy of our detection algorithm in different traffic scenarios.



		Interference channel										
		1	2	3	4	5	6	7	8	9	10	11
Channel setup of legitimate AP and PrAP	PrAP(ch1), LAP(ch6)	-	-	-	-	0.03	0.12	0.09	0.02	0.05	1	1
	PrAP(ch2), LAP(ch6)	-	-	-	-	-	0.03	0.03	0.03	0.05	1	1
	PrAP(ch3), LAP(ch6)	-	-	-	-	-	-	0.06	0.02	0.03	1	1
	PrAP(ch4), LAP(ch6)	-	-	-	-	-	-	-	0.03	0.02	1	1
	PrAP(ch5), LAP(ch6)	1	-	-	-	-	-	-	-	0.05	1	1
	PrAP(ch7), LAP(ch6)	1	1	0.03	-	-	-	-	-	-	-	1
	PrAP(ch8), LAP(ch6)	1	1	0.15	0.03	-	-	-	-	-	-	1
	PrAP(ch9), LAP(ch6)	1	1	0.05	0.03	0.01	-	-	-	-	-	-
	PrAP(ch10), LAP(ch6)	1	1	0.04	0.01	0.04	0.12	-	-	-	-	-
	PrAP(ch11), LAP(ch6)	1	1	0.13	0.00	0.10	0.11	0.06	-	-	-	-

Figure 16. Features of Φ_{ch} shown under an idle traffic scenario of the desktop PrAP-Hunter (500Kbps, 250FPS). RAP is the currently-connected AP, and it is relaying signals between a PrAP-Hunter and a legitimate AP (LAP).



(a) Distribution of Φ_{min}s

(b) Error rate against Φ

Figure 17. (a) Desktop PrAP-Hunter: distribution of Φ_{min}s; idle traffic. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) CDF of false negative and false positive rates against Φ.

the bandwidth limit of IEEE802.11n with MIMO (two antennas). We conducted experiments for 600 times with a PrAP under idle traffic. As a result, the proposed method only failed one time. We repeated our experiments with a legitimate AP for 600 times, and the proposed method successfully identified the legitimate AP without an error. Similar experiments were conducted in a heavy traffic scenario, and as a result, the method failed 10 times with the PrAP and 12 times with a legitimate AP. In the following, we examine the results of both scenarios in more details.

Results in an Idle Traffic Scenario. In an idle traffic scenario, we examined the proposed method against a PrAP with different channel combinations. Figure 16 shows the results in details. The first column shows the channel setup of the legitimate AP and the PrAP, and the first row lists interference channels (our interference device purposely interferes with the PrAP channel by sending beacons through a legitimate AP's channel.). To detect a PrAP, the PrAP-Hunter connected to the PrAP and it sent data. For simplicity, we only listed Φ_{ch}s of which interference channel ch had a gap of more than 3 channels from the PrAP's channel. As a result, we observed that all the interference channels of which Φ_{ch}s were less than our fixed threshold of 0.5 (from channel 3 to channel 9) shared bandwidth with channel 6 of the legitimate AP. That is, under the existence of a legitimate AP on channel 6, a PrAP will be caught by our algorithm irrespective of what channel the attacker chooses to use. As described in §4, when a PrAP relays traffic between a station and a legitimate AP, the throughput in both channels of the two APs contribute to data transmission. When interference signals are applied to channels that share bandwidth with a legitimate AP, we observe traffic obstruction from the standpoint of the PrAP-Hunter using an independent channel.

We collected all instances of Φ_{min} in each detection trial to analyze the distribution in idle traffic experiments. As shown in Figure 17(a), when we tested our algorithm with PrAP, most of the Φ_{min}s in each detection trial were less than 0.4. With a legitimate AP, all Φ_{min}s in each detection trial were greater than 0.87.

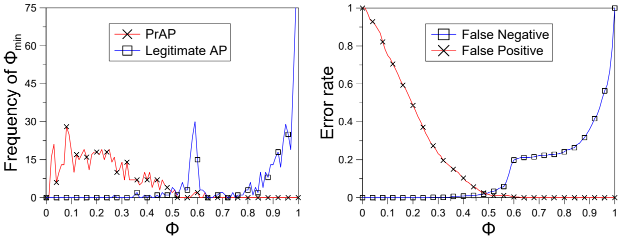
Figure 17(b) shows the legitimate AP's and PrAP's detection error rate against Φ. The detection threshold was between 0.54 and 0.87, which kept both false positive and false negative rates at 0%. Even though we used a fixed detection threshold at 0.5 to distinguish legitimate APs and PrAPs, we could obtain a false positive rate of 0% and a low false negative rate at less than 1%.

Results in a Heavy Traffic Scenario. Results in a heavy traffic scenario are almost identical to those in the idle scenario. Distribution in a heavy traffic case in Figure 18(a) looks more noisy than that in the idle case in Figure 17(a). However, as shown in Figure 18(a), for the PrAP, most Φ_{min}'s in each detection attempt were less than the fixed detection threshold of 0.5. With a legitimate AP, most Φ_{min}'s in each detection attempt were greater than 0.5. Figure 18(b) shows the legitimate AP and PrAP detection error rate against Φ. We observe that a detection threshold of 0.49–0.50 could keep both false positive and false negative rates less than 2%. In this paper, we used a fixed detection threshold at 0.5 to distinguish legitimate APs and PrAPs, which produced a sum of false positive and false negative rate of less than 3.67%.

7.3 Mobile PrAP-Hunter

We performed our experiments in an idle traffic scenario with the mobile PrAP-Hunter, since we only wanted to know whether our method could work well when the PrAP-Hunter is placed far from both the legitimate AP and the PrAP (c.f. §8 for the performance in a heavy traffic scenario). For that reason, we performed our

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.



(a) Distribution of Φ_{min}s (b) Error rate against Φ

Figure 18. (a) Desktop PrAP-Hunter: distribution of Φ_{min}s; heavy traffic. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) CDF of false negative and false positive rates against Φ.

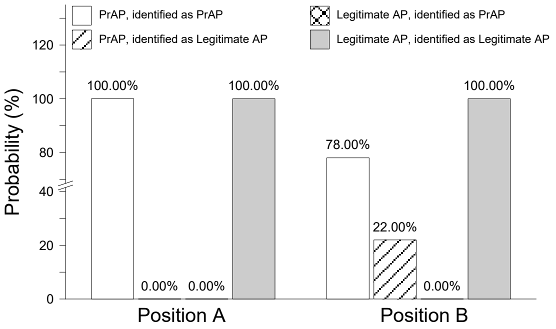
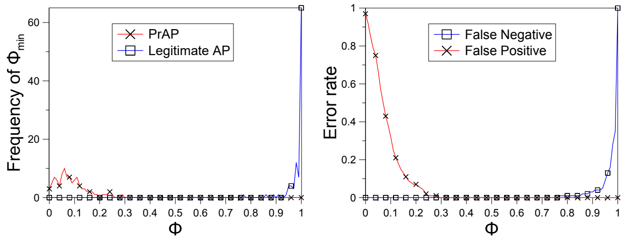


Figure 19. A plot examining the accuracy (measured by the true positive, true negative, false positive and false negative) of our detection algorithm at different positions.

experiments in two different positions: A and B, as shown in Figure 13. Figure 19 summarizes the results of our experiments using the mobile PrAP-Hunter. In each position, we examined the proposed method against both legitimate AP and PrAP 100 times, respectively. As a result, experiments in position A showed 100% success rate in detecting both legitimate AP and PrAP. In position B, the PrAP-Hunter failed 22 times against the PrAP and never failed against the legitimate AP.

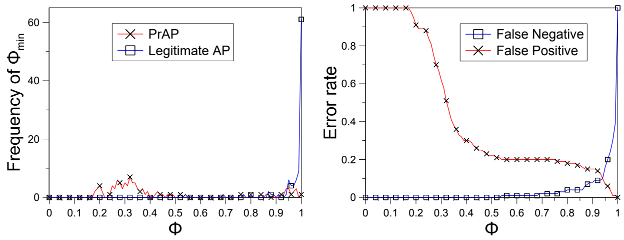
Results in Position A. The experiment setting was same as in the desktop PrAP-Hunter experiment. As shown in Figure 20(a), the distribution of the Φ_{min} was similar to the results shown in the idle traffic scenario of the desktop PrAP-Hunter experiments. Legitimate APs and PrAPs could clearly be distinguished, because all Φ_{min}s for each PrAP detection were less than 0.3, and for legitimate AP detection, they were greater than 0.85. Figure 20(b) shows the legitimate AP and the PrAP detection error rate against Φ. As shown in the figure, we can keep both false positive and false negative rates at 0% when we set the detection threshold between 0.3 and 0.78. Thus, when we use a fixed detection threshold at 0.5, our PrAP-Hunter produced a 100% success rate in both legitimate AP and PrAP detections.

Results in Position B. We repeated detection experiments 100 times for each legitimate AP and PrAP in position B, where the distance between them is 10 meters. Figure 21(a) shows the distribution of Φ_{min}, where the PrAP-Hunter was able to maintain a stable data transmission with the used AP. Thus, when we tested the proposed method with a legitimate AP, most of the Φ_{min} values were greater than 0.8. With the PrAP, most Φ_{min} values in each trial were less than 0.5, and greater than 0.8 only in a few cases; that happened only when the channel gap between the legitimate AP and the PrAP was only 1. For example, assume that a legitimate AP used channel 6 and a PrAP used channel 5. We should interfere only with channel 6, the legitimate AP's channel, but not the PrAP's channel (channel 5) to observe the traffic obstruction by the interference. To do so, channel 9 would be the best choice. When the interference device works on channel 9, it would interfere with the legitimate AP's channel (channel 6) via overlapping channels 6, 7, and 8 successfully, but not the PrAP's (channel 5). Unfortunately, signals of the interference device are attenuated significantly due to the distance. Thus, the number of channels affected by the interference device was only 5 (channels 7, 8, 9, 10, and 11. Not 7 channels as we expected in our detection strategy). That is, only channels 7 and 8 (but not channel 6) were affected, so beacon transmission over channel 9 did not successfully interfere with the legitimate AP's channel. This exceptional case happens only when: (1) the PrAP-Hunter is far both from legitimate AP and PrAP, and (2) the service channel gap is 1. However, we can break condition (1) by moving our PrAP-Hunter closer to an AP of interest using proper SNR values.



(a) Distribution of Φ_{min} (b) Error rate against Φ

Figure 20. Results. (a) Mobile PrAP-Hunter: the distribution of Φ_{min} at position A in our experimental setup. The detection trials were repeated 100 times for the rogue and legitimate AP measurements, respectively. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) The CDF of the false negative and false positive rates against various values of Φ.



(a) Distribution Φ_{min} (b) Error rate against the Φ

Figure 21. Results. (a) Mobile PrAP-Hunter: the distribution of Φ_{min} at position B, where detection trials were repeated 100 times for both the rogue and legitimate AP measurements, respectively. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) The CDF of the false negative and false positive rates against various values of Φ.

Remark on detection position. There was one extreme case where the PrAP-Hunter was out of the range of the legitimate AP, but within the range of the PrAP. Even in this case, the PrAP-Hunter still could interfere with the communicating channel between the legitimate AP and the PrAP, and it successfully detected the PrAP, because the PrAP's network interface was still within the PrAP-Hunter's range.

In summary, we conclude that PrAP-Hunter's position affects detection performance: a larger distance between PrAP-Hunter and APs caused a higher error rate. Generally, since a station will select an AP with the highest SNR, a PrAP should be located close to the station to allow connection. Also, a network administrator using our PrAP-Hunter can easily find the location close to an AP of interest using SNR values, and scenarios shown in position B can be easily avoided. Figure 22 summarizes our experiments.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

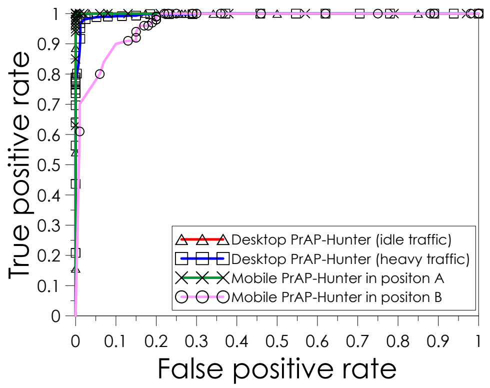


Figure 22. The Receiver Operating Characteristic (ROC) curve capturing the trade-off between the false and true positive rates, and various experimenting settings for both the desktop and mobile versions of PrAP-Hunter. The results show that one can choose an optimal threshold to achieve a high true positive without sacrificing the low false positive rate in most settings.

8 Detection In the Wild

To illustrate PrAP-Hunter in the real-world, we conducted experiments at coffee shops upon obtaining the store's permission and an approval from our institutional review board (IRB) assuring that our experiments are in no way going to harm users.

8.1 Hide-and-Seek Game

We designed a "hide-and-seek" game to show how our PrAP-Hunter performs in real world (10 different coffee shops). Below, we describe this game; settings, detection strategy and results.

Settings. For this game, we had two players: attacker (hider) and PrAP-Hunter (seeker). We designed and developed our hardware PrAP so that it was easily deployed in real world: it only needed a power source for operation with all parameters pre-defined and set. For our experiments, the attacker may (or may not) decide to deploy a PrAP in the tested environment. If he decides to deploy a PrAP, the PrAP was turned on and its position was determined by the attacker. For more realistic experiments, the location of the PrAP was chosen randomly. PrAP-Hunter (the defender) knew the location of the legitimate AP, since it was visible to users as well as PrAP-Hunter. However, PrAP-Hunter did not know the location of PrAP nor whether a PrAP was turned on or off. The PrAP-Hunter was assumed to automatically connect to the PrAP when it had the highest power signal in the deployment environment. We noticed that this assumption was reasonable: in all the stores where we ran our game, the default Wi-Fi manager did not allow choosing an SSID working on a specific channel, but rather automatically connected to the AP with the highest power.

Strategy. First, the PrAP-Hunter finds the position of the legitimate AP, which is visible and often located by the cashier as shown in Figure 24(a). Then, the PrAP-Hunter chooses a Wi-Fi connection position, and our choice of this position must ensure that the PrAP has a stronger signal than the legitimate AP's, so that a legitimate user may connect to the PrAP automatically. Accordingly, the Wi-Fi connection position must be far from the visible legitimate AP. Once connected, we start the detection phase.

Results. Based on the settings and strategy described above, the two players execute the game: one player hides the PrAP and the other tries to find it. The PrAP is turned either on or off by the hider, but the choice is not known to the seeker (PrAP-Hunter). After all set up, the seeker comes into the store, and tries to find whether a PrAP exists or not using our PrAP-Hunter. In the experiment, the detection rate was 100%, that is, the seeker correctly found 3 PrAPs and 7 legitimate APs at 10 different stores, which corresponds to the actual deployment of PrAPs. About 20-50 SSIDs were found in each store, and the experiments were conducted in the afternoon. The results are in Figure 23.

8.2 Understanding the Effect of Distance

To understand the effect of distance, we ran an experiment at a coffee shop, with the map shown in Figure 24(a). This cafe provided free WLAN with the same SSID at channel 1, 6, 11 and outlet services for customers. With these resources, we set up a PrAP as described in this paper. The PrAP relayed traffic of a legitimate AP serving on channel 1, and operated on channel 11 with the same SSID as a trusted AP. When we tested a legitimate AP using our PrAP-Hunter, we obtained results which were similar to Figure 5(a). If the AP on channel 11 was a legitimate AP, we should obtain results which were similar to the Figure 5(d). However, we obtained results which were similar to Figure 6(d), which means our PrAP-Hunter identified correctly the PrAP.

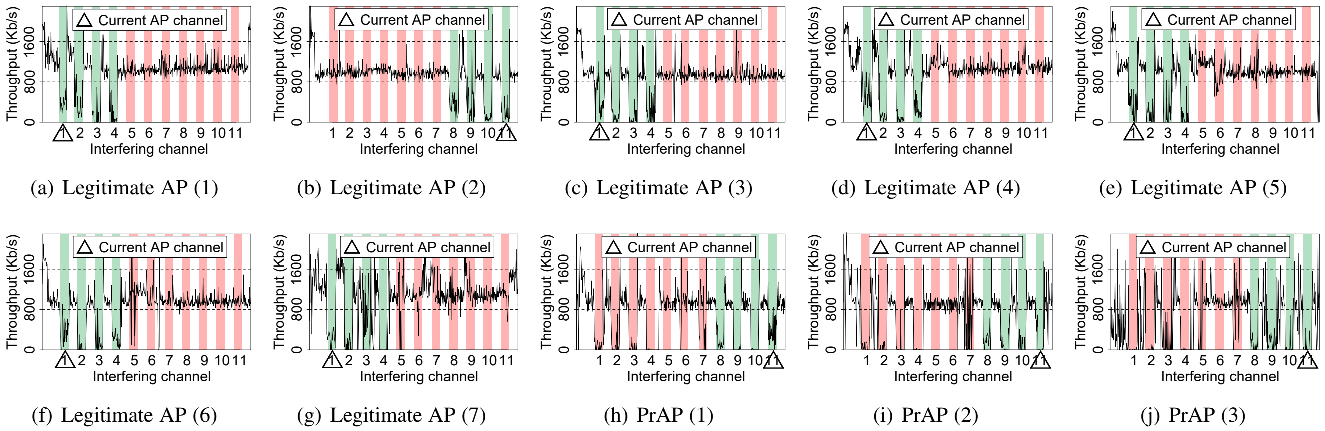
We performed our experiments in three different positions of the cafe. In position A, the PrAP-Hunter was close to the PrAP. In position B, the PrAP-Hunter had the same distance with both the legitimate AP and the PrAP. In position C, the PrAP-Hunter was close to the legitimate AP. Figure 24(b)-(d) shows results of PrAP detections in three positions. In the experiment, the PrAP-Hunter could not maintain a stable data transmission rate with the rogue AP. The reasons were as follows. First, we placed our traffic receiver in the intranet of our campus. When the PrAP-Hunter sent data from Internet to the intranet, the Internet traffic and dynamic routing paths led to unstable traffic transmission. Second, we performed our experiments at the peak time at the cafe, where another potential reason could be the high level of AP workload. Finally, we placed our PrAP in a backpack to hide it from people, so two wireless interfaces of the PrAP were placed too closely, which caused interference with each other in a small backpack, even though they used different channels [46]. As shown in Figure 24, although PrAP-Hunter showed unstable data transmission, we still could obtain good results that had similar features to Figure 6(d); the proposed method successfully found the existence of a PrAP even in various real world scenarios.

9 Discussion

A sophisticated rogue AP. We analyzed the security against a more sophisticated rogue AP that intentionally reduces the bandwidth of forwarding link when the PrAP-Hunter generates traffic, and increases the bandwidth when PrAP-Hunter generates interference. The derived degree of channel interference might be above the threshold, so the rogue AP might be able to avoid the detection in this case. We break down the sophisticated attacker's control of bandwidth into two cases: one is to increase the bandwidth while interfering, and the other is to decrease it while idle. For the increment of bandwidth, it does not affect our detection algorithm, because our interference is effective on the wireless channel. That is, even when the attacker increases the bandwidth while interfering by giving higher priority to the flow, it cannot increase the throughput for the given wireless channel. When decreasing the bandwidth, we note that our constant-rate traffic is already very low (only 500 Kbps in our experiments). Thus, even the lower transmission rate caused by the attacker

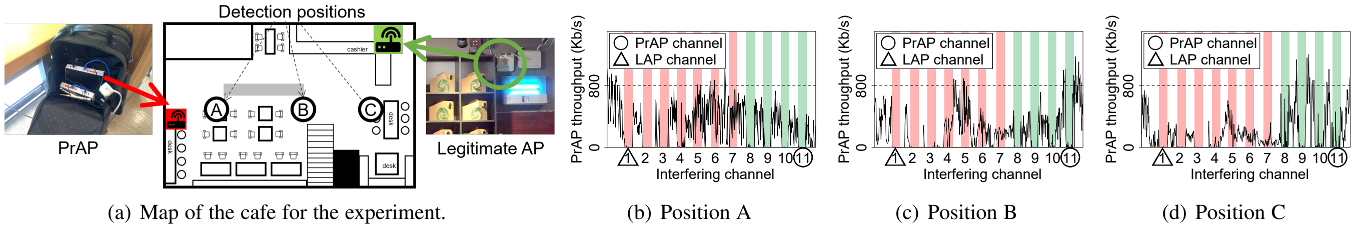
1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing



(a) Legitimate AP (1)
(b) Legitimate AP (2)
(c) Legitimate AP (3)
(d) Legitimate AP (4)
(e) Legitimate AP (5)
(f) Legitimate AP (6)
(g) Legitimate AP (7)
(h) PrAP (1)
(i) PrAP (2)
(j) PrAP (3)

Figure 23. Results of the hide-and-seek game at 10 coffee shops.



(a) Map of the cafe for the experiment.
(b) Position A
(c) Position B
(d) Position C

Figure 24. Legitimate AP serving in channel 1. Rogue AP relayed the Internet service of the legitimate AP and served as a trusted AP (using a cafe's wireless network SSID) in channel 11. In position A, the PrAP-Hunter was close to the PrAP. In B, the PrAP-Hunter had the same distance with both the legitimate AP and the PrAP. In C, the PrAP-Hunter was close to the legitimate AP.

can be a good indicator of the rogue APs existence. Finally, we note that to mount this attack, further investigation is required on whether an attacker is able to identify our detector's traffic or not to frustrate our detection algorithm: considering that the traffic generation and detection can be done in a random interval and duration, they will be mixed with normal users traffic.

Operation mode. Our design of PrAP-Hunter is generic, and is not limited to an after-fact deployment. When a rogue AP is detected using this algorithm, that same rogue AP might has already been used to successfully launch an array of attacks on innocent users connecting to it. To provide preventive countermeasure, we could use our system proactively and periodically to detect malicious and rogue access points as soon as they are deployed. We can set up our system to detect rogue APs in multiple fixed locations and to have them run the detection algorithm periodically, which will give us a high chance to detect rogue APs before they mount the attack.

Detection period. The more frequently the detector is operated, the worse the experience of the legitimate users would be due to interference, although the faster the detection is. This trade-off is a clear limitation of our approach. To balance this trade-off, our approach can be deployed over limited periods of time to detect the rogue APs. Furthermore, in order to address scenarios where the adversary would learn the operation cycles of our detector and try to avoid them by on/off operation, we can also envision that our system would operate by randomly hopping in the time domain for its operation, to be unpredictable.

Interference in 40MHz channels and 802.11ac. For wider channels in 2.4 GHz such as 40MHz, we note that 20MHz channel is most common, and 40MHz is hardly observed because the number of orthogonal channels is too small. Detecting in the 5GHz channels (as in IEEE802.11ac) is much easier with our advanced detection strategy, because channels do not share bandwidth between each other. Similarly, detecting a PrAP relaying 2.4GHz and 5GHz is easy with our strategy. We can interfere with one of the channels (either 2.4GHz or 5GHz channel) while sending data with the other channel to see whether it has two wireless channels or not.

3G/LTE channel. One possible system model scenario in which our attack would operate is a 3G/LTE channel used for relaying. In particular, one may assume that the attackers use such a 3G/LTE network to provide connectivity, by relying the rAP traffic of legitimate users, thus virtually violating the underlying assumption of our detector. In reality, however, in most of the case such an approach for relying traffic would still also be software-bridged, which would make detection even easier based on the time feature. Certainly, one can also perform a 3G/LTE-WiFi hardware bridge to avoid packet delay and eliminate the time feature used for the time-based detection. However, as a defense, one can also employ the same approach of WiFi jamming, utilized in our work for hardware-based rAP detection, but at the cellular network [47] to obtain similar detection results. Testing such a scenario experimentally is an orthogonal contribution to our work, and we will pursue that as a future study.

10 Conclusion

We introduced a PrAP that can evade the most widely advocated and used time-based detection techniques. We showed that while time-based techniques were indeed suitable for software-based rAP detection, they were obsolete against our new PrAP. Using various experiments, we showed the feasibility of our PrAP. To defend against its threat, we developed a new mechanism that used channel interference for PrAP detection. Our mechanism is capable of detecting hardware-based PrAPs, as demonstrated by various experimental scenarios and two deployment setups.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2903052, IEEE Transactions on Mobile Computing

Acknowledgments. This work was supported by the Global Research Lab. (GRL) Program of the National Research Foundation (NRF) funded by Ministry of Science, ICT (Information and Communication Technologies) and Future Planning (NRF-2016K1A1A2912757).

References

[1] R. Jang, J. Kang, A. Mohaisen, and D. Nyang, "Rogue access point detector using characteristics of channel overlapping in 802.11n," in *Proc. of IEEE International Conference on Distributed Computing Systems, ICDCS*, 2017, pp. 2515–2520.

[2] S. Brenza, A. Pawlowski, and C. Pöpper, "A practical investigation of identity theft vulnerabilities in eduroam," in *Proc. of ACM Conference on Security & Privacy in Wireless and Mobile Networks, S&P*, 2015, pp. 14:1–14:11.

[3] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A timing-based scheme for rogue AP detection," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 11, pp. 1912–1925, 2011.

[4] H. Gonzales, K. S. Bauer, J. Lindqvist, D. McCoy, and D. C. Sicker, "Practical defenses for evil twin attacks in 802.11," in *Proc. of IEEE Global Communications Conference, GLOBECOM*, 2010, pp. 1–6.

[5] C. Yang, Y. Song, and G. Gu, "Active user-side evil twin access point detection using statistical techniques," *IEEE Trans. Information Forensics and Security*, vol. 7, no. 5, pp. 1638–1651, 2012.

[6] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engel, "Undesired relatives: protection mechanisms against the evil twin attack in IEEE 802.11," in *Proc. of ACM Symposium on QoS and Security for Wireless and Mobile Networks, Q2SWinet*, 2014, pp. 87–94.

[7] ——, "Hacker's toolbox: Detecting software-based 802.11 evil twin access points," in *Proc. of IEEE Annual Consumer Communications and Networking Conference, CCNC*, 2015, pp. 225–232.

[8] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Pironti, and P. Strub, "Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS," in *Proc. of IEEE Symposium on Security and Privacy, S&P*, 2014, pp. 98–113.

[9] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "Sok: Security and privacy in implantable medical devices and body area networks," in *Proc. of IEEE Symposium on Security and Privacy, S&P*, 2014, pp. 524–539.

[10] A. M. Bates, J. Pletcher, T. Nichols, B. Hollembaek, D. Tian, K. R. B. Butler, and A. Alkhelaifi, "Securing SSL certificate verification through dynamic linking," in *Proc. of ACM SIGSAC Conference on Computer and Communications Security, SIGSAC*, 2014, pp. 394–405.

[11] N. Cheng, "Take precautions on public Wi-Fi," The Nation, August 2016.

[12] L. Constantin, "This tool can alert you about evil twin access points in the area," InfoWorld, April 2015.

[13] R. Gery, "Beware the 'evil' Wi-Fi networks that turn your phone into a brick: Hackers can hijack systems to remotely attack your handset," Dailymail, April 2015.

[14] A. Baxter, "How to stop hackers from stealing your information on public Wi-Fi," The Next Web, June 2015.

[15] E. Tuvey, "The dangers of using public Wi-Fi hotspots," betnews, March 2016.

[16] P. Cooper, "White hat hackers steal data from london wi-fi users in "evil twin" attack," Security News, November 2013.

[17] R. A. Beyah and A. Venkataraman, "Rogue-access-point detection: Challenges, solutions, and future directions," *IEEE Security & Privacy*, vol. 9, no. 5, pp. 56–61, 2011.

[18] N. Cheng, "Take precautions on public Wi-Fi," The Star, August 2016.

[19] A. Adya, P. Bahl, R. Chandra, and L. Qiu, "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks," in *Proc. of ACM Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2004, pp. 30–44.

[20] S. Nikbakhsh, A. B. A. Manaf, M. Zamani, and M. Janbeglou, "A novel approach for rogue access point detection on the client-side," in *Proc. of International Conference on Advanced Information Networking and Applications Workshops, WAINA*, 2012, pp. 684–687.

[21] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Enhancing the security of corporate Wi-Fi ntworks using DAIR," in *Proc. of ACM International Conference on Mobile Systems, Applications, and Services, MobiSys*, 2006, pp. 1–14.

[22] R. Chandra, J. Padhye, A. Wolman, and B. Zill, "A location-based management system for enterprise wireless LANs," in *Proc. of USENIX Symposium on Networked Systems Design and Implementation NSDI*, 2007.

[23] ——, "A location-based management system for enterprise wireless LANs," 2007.

[24] R. A. Beyah, S. Kangude, G. Yu, B. Strickland, and J. A. Copeland, "Rogue access point detection using temporal traffic characteristics," in *Proc. of IEEE Global Telecommunications Conference, GLOBECOM*, 2004, pp. 2271–2275.

[25] L. Watkins, R. A. Beyah, and C. L. Corbett, "A passive approach to rogue access point detection," in *Proc. of IEEE Global Communications Conference, GLOBECOM*, 2007, pp. 355–360.

[26] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. of ACM Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2008.

[27] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proc. of ACM Workshop on Wireless Security*, 2006, pp. 43–52.

[28] F. Lanze, A. Panchenko, B. Braatz, and T. Engel, "Letting the puss in boots sweat: detecting fake access points using dependency of clock skews on temperature," in *Proc. of ACM Symposium on Information, Computer and Communications Security, ASIACCS*, 2014, pp. 3–14.

[29] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 MAC layer spoofing using received signal strength," in *Proc. of the 27th Conference on Computer Communications, INFOCOM*, 2008.

[30] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *IEEE Trans. Mob. Comput.*, vol. 9, no. 3, pp. 449–462, 2010.

[31] W. Wei, S. Jaiswal, J. F. Kurose, and D. F. Towsley, "Identifying 802.11 traffic from passive measurements using iterative bayesian inference," in *Proc. of IEEE Conference on Computer Communications, INFOCOM*, 2006.

[32] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. F. Towsley, "Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs," in *Proc. of ACM SIGCOMM Internet Measurement Conference, IMC*, 2007, pp. 365–378.

[33] G. Qu and M. M. Nefcy, "Rapid: An indirect rogue access points detection system," in *Proc. of IEEE International Performance Computing and Communications Conference, IPCCC*, 2010, pp. 9–16.

[34] A. Venkataraman and R. A. Beyah, "Rogue access point detection using innate characteristics of the 802.11 MAC," in *Proc. of EAI International Conference Security and Privacy in Communication Networks, SecureComm*, 2009, pp. 394–416.

[35] C. D. Mano, A. Blaich, Q. Liao, Y. Jiang, D. A. Cieslak, D. Salyers, and A. Striegel, "RIPPS: rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 2, pp. 2:1–2:23, 2008.

[36] —, "Madwifi project," http://madwifi-project.org, November 2015.

[37] ——, "Aircrack-ng," http://www.aircrack-ng.org, November 2015.

[38] ——, "The karma software patch for access points," http://digi.ninja/karma, November 2015.

[39] T. Kindberg, J. Mitchell, J. Grimmett, C. Bevan, and E. O'Neill, "Authenticating public wireless networks with physical evidence," in *Proc. of the 5th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob*, 2009, pp. 394–399.

[40] V. Roth, W. Polak, E. G. Rieffel, and T. Turner, "Simple and effective defense against evil twin access points," in *Proc. of ACM Conference on Wireless Network Security, WISEC*, 2008, pp. 220–235.

[41] K. S. Bauer, H. Gonzales, and D. McCoy, "Mitigating evil twin attacks in 802.11," in *Proc. of IEEE International Performance, Computing and Communications Conference, IPCCC*, 2008, pp. 513–516.

[42] H. Gonzales, K. S. Bauer, J. Lindqvist, D. McCoy, and D. C. Sicker, "Practical defenses for evil twin attacks in 802.11," in *Proc. of IEEE Global Communications Conference, 2010. GLOBECOM*, 2010, pp. 1–6.

[43] J. Pang, B. Greenstein, M. Kaminsky, D. McCoy, and S. Seshan, "Wifi-reports: Improving wireless network selection with collaboration," *IEEE Trans. Mob. Comput.*, vol. 9, no. 12, pp. 1713–1731, 2010.

[44] Q. Wang, P. Xu, K. Ren, and X. Li, "Towards optimal adaptive UFH-based anti-jamming wireless communication," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 16–30, 2012.

[45] E. G. Villegas, E. López-Aguilera, R. Vidal, and J. Paradells, "Effect of adjacent-channel interference in IEEE 802.11 WLANs," in *Proc. of IEEE International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications, CROWNCOM*, 2007, pp. 118–125.

[46] A. Zubow and R. Sombrutzki, "Adjacent channel interference in IEEE 802.11n," in *Proc. of IEEE Wireless Communications and Networking Conference, WCNC*, 2012, pp. 1163–1168.

[47] M. Lichtman, J. H. Reed, T. C. Clancy, and M. Norton, "Vulnerability of LTE to hostile interference," in *Proc. of IEEE Global Conference on Signal and Information Processing, GlobalSIP*, 2013, pp. 285–288.

1536-1233 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.