Rogue Access Point Detector Using Characteristics of Channel Overlapping in 802.11n

RhongHo Jang†, Jeonil Kang†, Aziz Mohaisen‡, DaeHun Nyang†
†The School of Computer and Information Engineering, Inha University, Korea
‡Department of Computer Science and Engineering, University at Buffalo, USA
Email: {jiyoo, dreamx}@seclab.inha.ac.kr, amohaisen@gmail.com, nyang@inha.ac.kr

***Abstract*—In this work, we introduce a powerful hardware-based rogue access point (PrAP), which can relay traffic between a legitimate AP and a wireless station back and forth, 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 extensive experiments, we demonstrate that the state-of-the-art time-based rogue AP (rAP) detectors cannot detect our PrAP, although effective against software-based rAP. To defend against PrAPs, we propose PrAP-Hunter based on intentional channel interference. PrAP-Hunter is highly accurate, even under heavy traffic scenarios. Using a high-performance (desktop) and low-performance (mobile) 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 PrAP-Hunter is fast (takes 5-10 sec), does not require any prior knowledge, and can be deployed in the wild by real world experiments at 10 coffee shops.**

Keywords. Intrusion detection, Wireless LAN, Rogue AP, channel interference, IEEE 802.11n.

I. Introduction

With many public spaces, such as shopping malls, restaurants, and public transit systems providing WLAN services and power outlets for customers, an adversary with a laptop and an additional network interface can easily create a *persistent* rogue access point (rAP) to eavesdrop on, intercept, or even modify communications between users and the Internet (Figure 1). Such an adversary can use rAP 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.

In this work, we unveil limitations of the time-based rAP detection techniques by demonstrating that the delay used in the literature for inferring whether a rAPs exists between a user and a legitimate AP is not the result of an additional wireless path, but rather the result of a computational delay caused by the *software bridging*. To further show that is the case, 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 a detector that relies on new assumptions, and demonstrate its effectiveness in detecting the proposed powerful hardware-based rAP (PrAP), which we call PrAP-Hunter (see figures 8 and 9). Our solution achieved close to 100% detection rate.

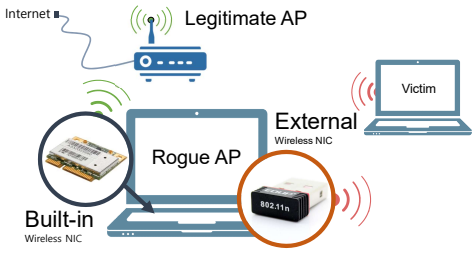


Fig. 1. General rouge Access Point (rAP) setup using a laptop with a built-in and an external wireless interfaces.

II. Related work

Time-based rAP detection 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.

Beyah *et al.* suggested a method that utilizes temporal characteristics, such as inter-packet arrival time [1]. Similarly, Yang *et al.* proposed an "evil twin" detector using a discriminative feature of inter-packet arrival time of a rAP [6]. Wei *et al.* [4], [5] proposed two similar detection schemes by examining the arrival time of consecutive ACK pairs in TCP traffic. Han *et al.* developed a software-based detection technique that uses round trip time (RTTs) of wire and wireless lines [3]. All of those techniques use packet delay of traffic caused by the rAP as a feature for detection.

III. MOVTIVATION

In many time-based rAP detection methods, researchers stated that when packets were sent through rAPs, packet delay would occur because rAPs used an additional wireless path.

However, 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 implemented a time-based detector described by Han

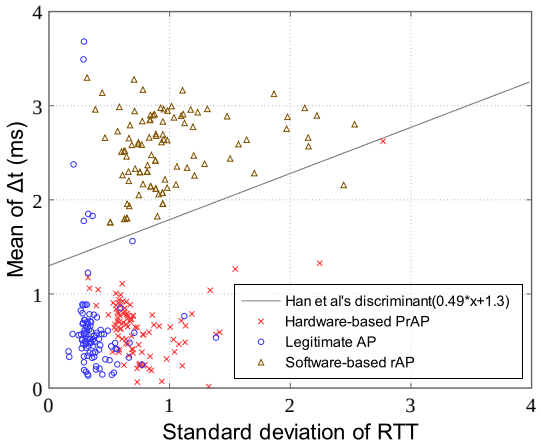


Fig. 2. Results of Han *et al.*'s [3] algorithm for two different rogue APs. One is a software-based rAP, and the other is a hardware-based PrAP.

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. Then, we performed experiments for Han *et al.*'s algorithm under the rAP and the PrAP described as follows.

1) Software-based rAP: In the literature, rAPs are defined using a laptop and an additional WLAN USB adapter, as shown in Figure 1 [3], [6]. 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.

2) Hardware-based PrAP: Figure 3 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 be detected using time-based rAP detection methods. Thus, we only consider an attacker using a hardware-based PrAP to forward packets with minimal delay to avoid time-based detectors in our threat model.

Figure 2 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), which supports our conclusion).

IV. PrAP Detection Strategy

A. The Basic Concept

Our PrAP-Hunter has two wireless interfaces, one that associates itself with a target AP to generate traffic to a receiver during the detection process, while the second interface (interference device) interferes with channel 1 to 11 sequentially with a rest time. Figure 4 illustrates how the proposed method works. The PrAP-Hunter connects to the PrAP (ch 11), which relays signals between the legitimate AP (ch 1) and a PrAP-Hunter (ch 11). When PrAP-Hunter generates traffic to the receiver, both channel 1 and channel 11 contribute to the data transmission. From the standpoint of the PrAP-Hunter, obstruction of data transmission is observed

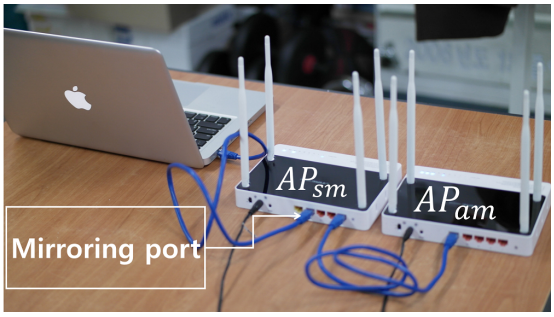


Fig. 3. Hardware setting of the PrAP (EFM ipTIME N8004R). AP_{sm} is responsible for repeating signals to and from the legitimate AP. AP_{am} and AP_{sm} are interconnected with a LAN cable via a wireless bridge function, and AP_{am} is assigned a valid IP from a DHCP server of AP_{sm} with a spoofed 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 the devices operated in the IEEE 802.11n mode with MIMO.

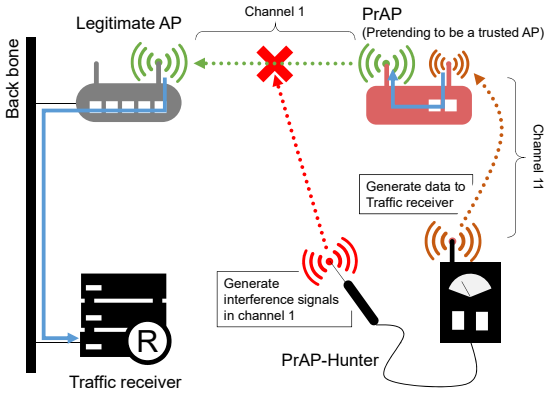


Fig. 4. A legitimate AP on channel 1 and a PrAP repeating signals of the legitimate AP on channel 11. The PrAP-Hunter generates traffic to the traffic receiver through the PrAP. The interference device interferes with channel 1. An AP is said to be rogue if we observe obstruction of traffic on channel 11 via the PrAP-Hunter.

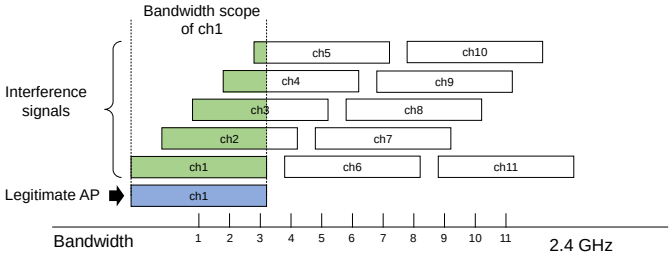
at channel 11 when the interference device interferes with channel 1 in order to determine whether the connected AP relays the signals. When that happens, the connected AP must be a PrAP.

B. Channel Interference in 802.11n

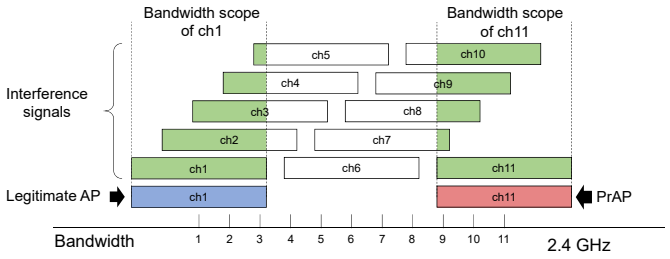
The channels used for WLAN are separated by 5 MHz in most cases and have a bandwidth of 20 MHz, per the 802.11n standard. In other words, each channel shares bandwidth with other adjacent channels. Considering a 20 MHz bandwidth channel, there is 17 MHz of bandwidth shared between channels 1 and 2, and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

C. Advanced Detection Strategy

Figure 5 shows our PrAP detection strategy, considering the wireless bandwidth standpoint. In Figure 5(a), we show a detection scenario where the legitimate AP uses channel 1 and no PrAP exists. PrAP-Hunter generates traffic through



(a) With legitimate AP



(b) With PrAP

Fig. 5. Channel interference under IEEE 802.11n

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 6(a). Channel 5 also shares a 2 MHz bandwidth with channel 1, but 2 MHz bandwidth sharing is not enough to interfere substantially. Also, based on both the work in [7] and our experiments, if PrAP-Hunter and the interference device are located farther than 50 cm apart, channel interference caused by 2 MHz bandwidth sharing is insignificant. As a result, we obtained Figure 6(a). Throughput degradation for the other channels by the interference are shown in Figure 6, 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, respectively. Figure 5(b) shows a detection scenario where a PrAP (ch 11) repeats a signal of legitimate AP (ch 1). If the AP connected on channel 11 was legitimate, the results of detection should look similar to the results reported in Figure 6(d). However, because we experienced an unexpected throughput degradation on channel 11 as shown in Figure 7(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, providing wireless connectivity by repeating signals.

In other words, if the number of obstructed channels is more than that of the legitimate AP's only scenario (that is, if the number of throughput degradation in Figure 7 is greater than that in Figure 6), there must be a PrAP in the system.

1) Degree of throughput degradation: Before each channel interference, 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 throughput degradation Φ as $\Phi_{ch} = \mathsf{itm}_{ch}/\mathsf{ntm}_{ch}$. In practice, we only measure Φ_{ch}s that $|ch - \mathsf{ch}_{ap}| > 3$ (Red bar channels in Figure 6 and Figure 7) and denote the minimum of Φ_{ch}s as Φ_{min} .

2) Time of Detection: In our experiments, we measure each Φ_{ch} in 5 sec epochs (3 sec for traffic recovery and 2 sec for interfering) for 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 Figure 6, we do not need to interfere with all channels but with only 2 channels. Thus, we spend 5 sec at a minimum and 10 sec at a maximum.

3) Interference message: We need a message that contained large amounts of data to stably generate interference signals. Also, messages should be broadcast to all devices, because which legitimate AP is used by the PrAP is not known in advance. Thus, we use a beacon frame which the size is modified to contain up to 1500 bytes. For sizing up our beacon frame, random information is added in the network data field.

4) Limitations: Although our scheme can detect advanced PrAPs using close channels to a legitimate AP, it cannot effectively detect PrAPs that use the same channel as the legitimate AP. Fine control of throughput degradation and interference degree is expected to overcome this limitation of our work. However, we believe in the opportunities of furthering this research, and leave as a future work for further investigation.

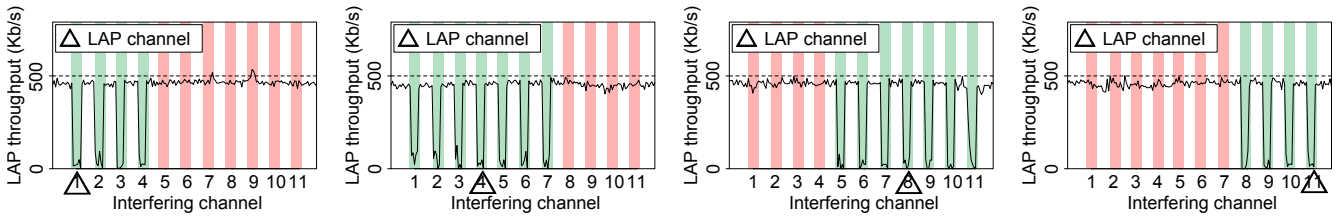
V. PrAP-Hunter Setup

We implemented our PrAP-Hunter in two settings: a desktop computer setting and a mobile phone setting. The first PrAP-Hunter was implemented on a high-end hardware in a fixed position for analyzing the performance under various traffic scenarios. The mobile PrAP-Hunter was implemented on a relatively low-performance mobile device, and is used for analyzing the performance in the wild.

A. Desktop Detector

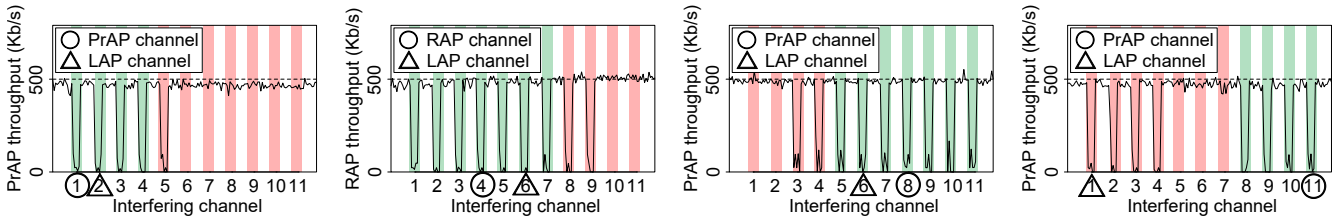
The hardware configuration of our desktop PrAP-Hunter is a box equipped with an Intel Core i5-3570K CPU, 4GB RAM, an ipTIME n500U external wireless card as a traffic generator, and a D-Link DWA-125 external wireless card as an interference device (see Figure 8 for a visual demonstration).

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 [7], the distance between devices is



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

Fig. 6. 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 on channel 1 and legitimate AP on channel 2 (b) PrAP on channel 4 and legitimate AP on channel 6 (c) PrAP on channel 8 and legitimate AP on channel 6 (d) PrAP on channel 11 and legitimate AP on channel 1

Fig. 7. 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).

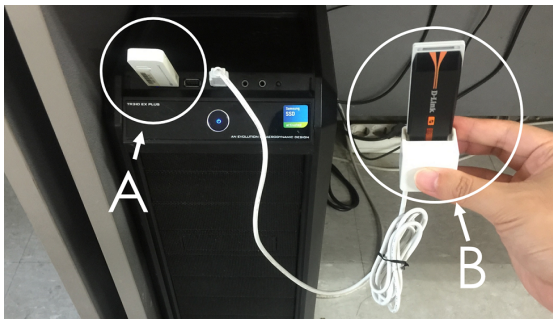


Fig. 8. Hardware setting of the desktop PrAP-Hunter. A is an wireless interface that is connected to the target AP and generates traffic, and B is the wireless interface that sends interference signals through 2.4GHz channels.

Fig. 9. Hardware setting of the mobile PrAP-Hunter.

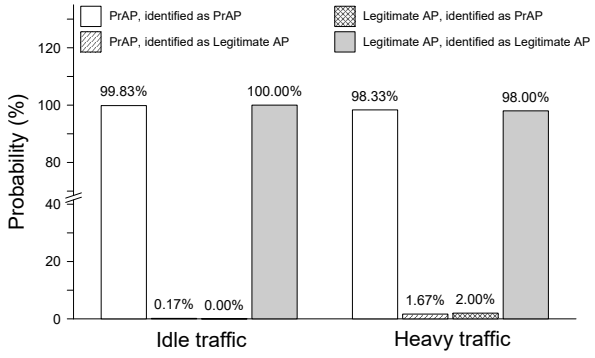


Fig. 10. Examining the accuracy of our detection algorithm in different traffic scenarios.

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.

B. Mobile PrAP-Hunter

Figure 9 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 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.

VI. Performance

A. Desktop PrAP-Hunter

Figure 10 summarizes the results of our experiments in an idle and a heavy traffic scenarios. For the idle traffic

Channel setup of legitimate AP and PrAP		Interference channel										
		1	2	3	4	5	6	7	8	9	10	11
	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	-	-	-	-

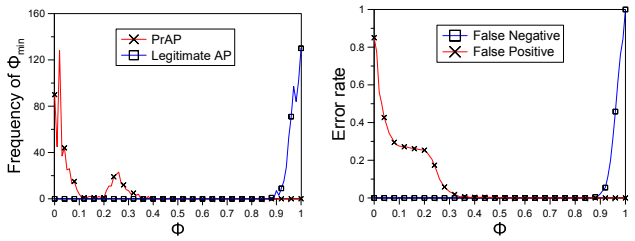
Fig. 11. Features of Φ_{ch} shown under an idle traffic scenario of the desktop PrAP-Hunter (500Kbps, 250FPS). RAP is the currently-connected AP, and it is relaying signals between a PrAP-Hunter and a legitimate AP (LAP).

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 the bandwidth limit of IEEE802.11n with MIMO (two antennas).

We conducted experiments for 600 times with a PrAP using the proposed method 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. 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 details.

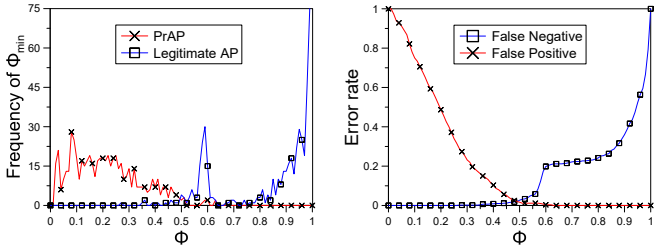
1) Results in an Idle Traffic Scenario: In an idle traffic scenario, we examined the proposed method against a PrAP with different channel combinations. Figure 11 shows the results in details. The first column shows the channel setup of the legitimate AP and the PrAP, and the first row lists interference 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.

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 §IV, 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.



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

Fig. 12. (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.



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

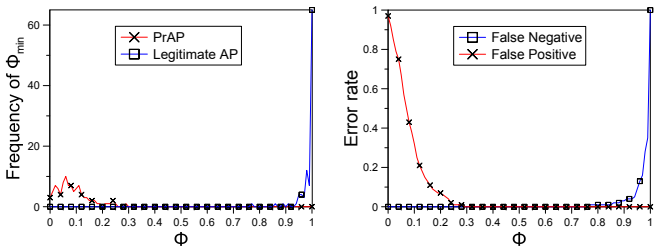
Fig. 13. (a) Desktop PrAP-Hunter: distribution of Φ_{min}s; heavy traffic. y-axis ($= f(\Phi_{min})$) shows the frequency of Φ_{min}. (b) CDF of false negative and false positive rates against Φ.

We collected all instances of Φ_{min} in each trial to analyze the distribution in idle settings. As shown in Figure 12(a), when we tested our algorithm with a 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 12(b) shows the legitimate AP's and PrAP's detection error rate against Φ. The detection threshold of 0.54 to 0.87 could keep both false positive and false negative rates at 0%.

2) 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 13(a) looks more noisy than that in an idle case in Figure 12(a). However, as shown in Figure 13(a), for the PrAP, most of the Φ_{min}'s in each detection attempt were less than 0.5. With a legitimate AP, most of the Φ_{min}'s in each detection attempt were greater than 0.5. Figure 13(b) shows that a detection threshold of 0.49 to 0.50 could keep both false positive and false negative rates less than 2%.

B. Mobile PrAP-Hunter

The experiment setting was same as in the desktop PrAP-Hunter experiment. We examined the proposed method against both legitimate AP and PrAP 100 times, respectively. As a result, experiments showed 100% success rate in detecting both legitimate AP and PrAP. As shown in Figure 14(a), 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 14(b) shows the legitimate AP and the PrAP detection



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

Fig. 14. Results showing the relationship between error rate and Φ. (a) Mobile PrAP-Hunter: distribution of Φ_{min}s. 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.

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.

VII. Detection In the Wild

To demonstrate our PrAP-Hunter in real world, we conducted real experiments at coffee shops. For that, we obtained permission from the store and an approval from our institutional review board (IRB). All of the the experiments are done in South Korea, and the IRB approval is obtained at Inha University assuring that our experiments are in no way going to harm users.

A. Hide-and-Seek Game

A "hide-and-seek" game to show how PrAP-Hunter performs in real world is designed and tested.

1) Settings: For this game, we had two players: attacker (hider) and PrAP-Hunter (seeker). We designed and developed our hardware PrAP so that it is easily deployed in practice: 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. PrAP-Hunter was assumed to automatically connect to PrAP when it had the highest power signal in the deployment environment. We note that this assumption is 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.

2) Strategy: We follow the following strategy. First, the PrAP-Hunter finds the position of the legitimate AP, which is visible and often located by the cashier 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.

3) 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.

VIII. Conclusion

We introduced and demonstrate 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.

Acknowledgement

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. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland, "Rogue access point detection using temporal traffic characteristics," in *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, vol. 4. IEEE, 2004, pp. 2271–2275.

[2] S. Brenza, A. Pawlowski, and C. Pöpper, "A practical investigation of identity theft vulnerabilities in eduroam," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2015, p. 14.

[3] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A timing-based scheme for rogue ap detection," *IEEE Transactions on parallel and distributed Systems*, vol. 22, no. 11, pp. 1912–1925, 2011.

[4] W. Wei, S. Jaiswal, J. Kurose, D. Towsley, K. Suh, and B. Wang, "Identifying 802.11 traffic from passive measurements using iterative bayesian inference," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 325–338, 2012. [Online]. Available: http://dx.doi.org/10.1109/TNET.2011.2159990

[5] 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 *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference, IMC 2007, San Diego, California, USA, October 24-26, 2007*, 2007, pp. 365–378. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298357

[6] 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. [Online]. Available: http://dx.doi.org/10.1109/TIFS.2012.2207383

[7] A. Zubow and R. Sombrutzki, "Adjacent channel interference in IEEE 802.11n," in *2012 IEEE Wireless Communications and Networking Conference, WCNC 2012, Paris, France, April 1-4, 2012*, 2012, pp. 1163–1168. [Online]. Available: http://dx.doi.org/10.1109/WCNC.2012.6213952