Wireless Spreading of WiFi APs Infections using WPS Flaws: an Epidemiological and Experimental Study

Amirali Sanatinia
College of Computer and Information Science
Northeastern University
Email: amirali@ccs.neu.edu

Sashank Narain
College of Computer and Information Science
Northeastern University
Email: sashank@ccs.neu.edu

Guevara Noubir
College of Computer and Information Science
Northeastern University
Email: noubir@ccs.neu.edu

***Abstract*—WiFi Access Points (APs) are ideal targets of attack. They have access to home internal networks which allows an adversary to easily carry out man-in-the-middle attacks and spread infections wirelessly. They can also be used to launch massive denial of service attacks that target the physical infrastructure as well as the RF spectrum (both WiFi and cellular). While Wired Equivalent Privacy (WEP) vulnerabilities are common knowledge, the flaws of the WiFi Protected Setup (WPS) protocol are less known. In this paper, we use an epidemiological approach, combined with experimental war-driving measurements to investigate the speed of infections spreading in four neighborhoods of Boston, MA, USA, with distinct population and demographics. Our analysis and experimental data indicate that such attacks are feasible. While the graph of WEP APs and WPS APs may not be fully connected, the combined graph of WEP-WPS APs is fully connected, making large scale spreading of infections feasible. Due to the unique characteristics of WPS, the absence of automated firmware upgrades and mechanisms to safely configure and administer APs; these attacks pose a significant threat that require serious attention and countermeasures to provide safe management of APs and their policies.**

I. Introduction

WiFi APs are ideal targets of attack. WiFi APs are the main gateway to residential networks and have direct access to home computers, as such they can easily be used by an adversary to carry out man-in-the-middle attacks. They also have access to un-encrypted traffic. They are also wirelessly interconnected, independent of the wired Internet. WiFi APs can transmit at one Watt, do not run anti-virus software, do not have an automatic updating mechanism, and are rarely patched by their users, thus enabling a wireless spreading of APs infections. This makes the detection and mitigation of their spreading very difficult. Furthermore, a compromise of WiFi APs can result in global scale denial-of-service attacks on both targeted remote Internet infrastructure and the RF spectrum. Due to the increasing trend of mobile operators offloading traffic from the cellular networks to WiFi networks [1], the WiFi RF spectrum (2.4GHz and 5.2GHz) is now coupled with the cellular bands. Jamming the WiFi would cascade in a collapse of cellular networks.

Today's APs have several documented vulnerabilities that make it plausible for large-scale compromise. Two recent attack vectors include DNS Rebinding as demonstrated by Craig Heffner in 2010 [2], and the Wi-Fi Protected Setup (WPS) vulnerability as demonstrated in 2011 (US-CERT Vulnerability Note VU#723755). Millions of wireless routers are susceptible to a variant of DNS rebinding attack [2]. This attack allows an adversary to interact with the wireless router's internal interface when a user visits a malicious website. This enables reprogramming numerous routers that have a default or easy to guess admin password. Such APs could then be used to initiate the spreading of an airborne APs infection. It has been proven that one can carry an efficient brute force attack on the PIN of the WPS protocol in 3 to 6 hours [3]. APs from major vendors have been affected. These include Cisco, Belkin, D-Link, Linksys, and Netgear. APs' embedded operating systems, such as VxWork and dd-wrt, are also known to suffer from vulnerabilities [4]. Such vulnerabilities show that the compromise of APs can spread wirelessly.

In this paper, we investigate the state of AP infections spreading in 2013 in four distinct neighborhoods of the city of Boston. Our investigation combines an analytical and experimental evaluation of the potential spreading of AP compromise using the recently discovered WPS flaw and relies on war-driving measurements carried out in March and May 2013. Our contribution is as follows:

- We developed a set of hardware / software tools and obtained war-driving measurements to assess the demographic of vulnerable APs, their communication range and connectivity, in four neighborhoods of Boston. We passively collected data from more than eighty-nine thousand APs. We found, surprisingly, that a large number of APs, 4901 (over 10%) still use WEP and many have WPS enabled, 25226 (38%). While the four investigated neighborhoods have different population demographics, they still exhibit similar AP characteristics.
- We used the SIR compartmental approach used in epidemiology to model the spreading of infections through WEP / WPS and admin password guessing. We applied the resulting model to our measurements data for various parameters.
- Our analysis, combined with experimental data indicate that while it is difficult for WPS or WEP to enable a massive wireless spreading of infections, the combined graph of WEP and WPS can result in a single connected component allowing a full spreading to all susceptible APs within 63 days, even with

978-1-4799-0895-0/13/$31.00 ©2013 IEEE

conservative parameters.

- We propose new designs and frameworks for APs to allow more scalable and flexible administration and configuration and increased interoperability.

The rest of this paper is organized as follows: In Section II, we present WPS and its design flaw, as well as an overview of two compartmental epidemiological models, SIR and SEIR. In Section III, we describe the threat model. Section IV explains our methodology, tools, neighborhood choices, and data collection process. In Section V, we present our analysis and results including connectivity and speed of spread. We then discuss possible countermeasures and related work in Section VI.

II. Background

A. WiFi Protected Setup (WPS)

WPS is a protocol that was designed and introduced in 2007 by the Wi-Fi Alliance to enable secure pairing of WiFi devices with compatible APs. WPS defines two primary methods, the PIN-based method and the Push Button Configuration (PBC) in which a physical button needs to be pushed on the AP. The PIN-based method is mandatory for all the WPS certified devices, while PBC is optional for the wireless clients but mandatory for the APs. WPS defines two additional out of band methods based on NFC and USB, which are less popular and not covered by the WPS certification. In December 2011, Stefan Viehböck discovered a major security flaw in the WPS protocol design [3]. This design flaw makes a brute force attack on the APs practical.

WPS runs as a series of messages within the Extensible Authentication Protocol (EAP) authentication framework. The core of WPS consists of the client and AP engaging in an eight step message-exchange. The eight messages exchanged are denoted by M1 to M8 in Figure 1. The PIN is an eight digit number that is divided into two four digit halves. The last digit in the second half is a checksum that reduces the effective size of the second half to 3 digits. The PIN has a key size of 10^7, but because of the way WPS is designed the key is split in two halves and each part is used in the authentication separately. If any half of the PIN is incorrect the AP will reply with an EAP-NACK message. If an adversary receives EAP-NACK after sending M4, they know the first half was incorrect and if they receive EAP-NACK after sending M7, they know the second half was incorrect. This would reduce the key size to $10^4 + 10^3$. Implementing a lockout mechanism after a certain number of unsuccessful attempts or introducing a delay is not part of the WPS specification. As such, majority of vendors have not implemented any preventive measures to mitigate a brute force attack.

WPS has two security problems; the first an innate design flaw (a protocol that relies on two separate weak keys of size 10^4 and 10^3), and the second an implementation and design flaw that does not provide means to mitigate or prevent brute force attacks. Although few vendors and models do provide a lockout mechanism after multiple unsuccessful attempts and a delayed retry mechanism, for most APs the brute force attack speed is only limited by the time needed by the AP to calculate

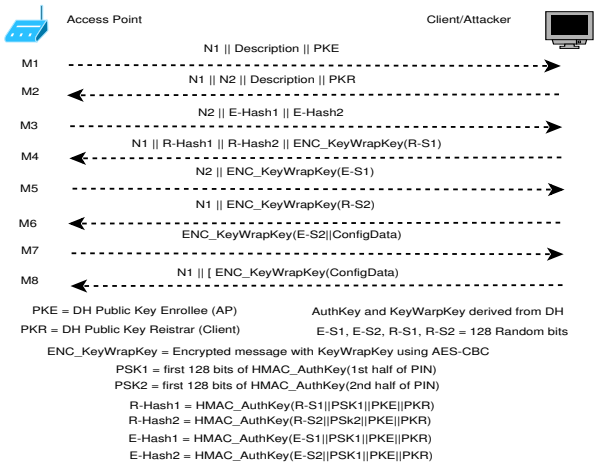


Fig. 1: Simplified diagram of WPS protocol, Phase1, Phase 2 and the authenticator block at the end of each message M have been omitted.

the Diffie-Hellman (DH) shared key (as mentioned in [3], using a small DH public key results in a significant speed-up). We experimented with 25 APs from 12 manufacturers and only two had a lockout mechanism.

Exploiting the WPS flaws has been made very practical by two tools; `reaver` and `wpscrack`. While `wpscrack` is a proof of concept implementation written in Python by Stefan Viehböck, `reaver` is written in C and is designed to be robust and practical against WPS, and is an active project already tested with a wide variety of APs. It provides numerous advanced options that gives the users a fine grain control and can be used to speed-up the process and increase the success rate.

B. Epidemiological Models

Modeling and analyzing the spread of computer viruses using epidemiological methods has been suggested and explored since the early days of Internet malware [5]. Compartmental models are one of the most frequently used frameworks to analyze the spread of an epidemic in a large population. These models reduce the diversity of the population to a few key characteristics that are relevant to the spread of the epidemic. This process results in subdivision of the population into compartments. The SIR model is a popular compartmental epidemiological model that uses three states: S (susceptible), I (infectious) and R (recovered). Each subject can only be in one of these three states [6], [7]. SIR is a simple model with good predictive power, suitable for modeling spread in biological systems, computer networks and scale-free networks [8], [9].

SEIR is an alternative model with four states: S (susceptible), E (Exposed), I (infectious) and R (recovered). It has been successfully used to study a specific class of biological infections [7]. In scenarios where there is a period of time, during which an individual has been infected but is not yet infectious, the SEIR model is more powerful and better suited. In this study, we use SIR model since it captures the characteristics of our network.

III. Threat Model

The design flaws in WPS makes it feasible for an adversary to compromise APs in a short duration of time. There are a number of trivial as well as sophisticated attacks that an adversary can implement to cause disruption. With just the Wi-Fi Protected Access (WPA) / WPA2 password, an adversary can eavesdrop on the unencrypted communication over the network, perform a man-in-the-middle attack or strip the TLS communications and harvest all credentials from the network. We look at the feasibility of a more complex and sophisticated attack by an adversary. After an AP's WPS PIN is compromised, an adversary can perform a dictionary attack to obtain the administrative credentials. The adversary can then re-flash the AP with malicious firmware. Our study indicates that not only is it easy to compromise the AP using the WPS design flaw, it is also easy to perform a dictionary attack to harvest the administrator credentials and re-flash the AP with a custom Linux firmware such as OpenWrt or DD-WRT which are based on the Linux kernel. OpenWrt and DD-WRT support a large database of APs and projects, such as our Open Infrastructure [10]. The Open Infrastructure project has made custom build images for a large variety of APs. Some vendors, such as Buffalo, ship their APs pre-installed with DD-WRT.

These compromised APs open up an array of opportunities for an adversary to conduct a wide range of attacks. It is trivial to implement man-in-the-middle, phishing and DNS pharming attacks since now the adversary just needs to change certain configurations of the firmware. These attacks have been discussed in [11]. The impact from the installation of custom firmware can be much more disruptive. For example, Fahl et al. recently studied over 13,500 popular free Android Apps from Google Play and found that 8.0% of the apps that use SSL/TLS were vulnerable to man-in-the-middle attacks. They demonstrated that an adversary can disable or even delete the antivirus application on Android by injecting "crafted" malware signatures that actually are the signature of the antivirus itself. A similar technique can be used to prevent or delay updates of desktop and mobile operating systems to make them vulnerable to malware and traditional attacks. Injection of malware on computers on the network can enable an adversary to gain complete control over a computer and use it in many ways, such as to harvest user credentials, access the resources on the computer, or use it as a bot in a distributed attack. Cui et al. [12] demonstrate how malicious firmware can be injected into embedded devices such as printers.

These compromised APs are also a threat to an individual's privacy. For example, the firmware can be setup to analyze all traffic that it sees, to infer information from unencrypted as well as encrypted traffic. Using techniques described in [13] and [14], it may be possible for an adversary to track encrypted Skype and VoIP communication. Even the contents of the calls can be leaked, such as the language of conversation. The firmware can also be setup to infer sensitive information about users on the network. Using the signal fluctuations, it may be possible to infer a user's whereabouts and movements and in conjunction with the list of registered devices, it may be possible for an adversary to find out when an apartment is unoccupied and rob it. It may also be possible for an adversary with access to numerous compromised APs to track individuals using their mobile device's MAC address. The possibility of the impact of such a scenario has been investigated in [15]. These APs can also practically be used as a part of a botnet to perform large-scale effective DDoS attacks. For example, PsyB0t, a worm that infected wireless APs and high-speed modems had been used to run a large number of zombies and run DDoS attacks against DroneBL.

In this paper, we focus on the spreading of infections across WPS / WEP APs in four specific neighborhoods of Boston. The spreading is limited by experimental data of location, range, and features of APs, as well as the failure probability of gaining administrative credentials to the APs or breaking the WPS PIN or WEP keys.

IV. Methodology

To study the spread of malware using connected wireless APs, we collected data from four distinct neighborhoods of Boston, MA, USA. The neighborhoods we chose for our study have different population demographics and urban architectures. The neighborhoods selected were such that we could study and analyze the similarities and differences in their wireless topology. To collect and analyze data for our study, we developed both software and hardware tools / systems. These tools helped us collect and analyze the data in a manner that addresses the privacy concerns of the users connected to these wireless networks.

A. Data Collection

The data was collected from four neighborhoods in Boston: Allston, Back Bay, Fenway and South Boston.

Allston is a residential neighborhood in Boston that is mostly populated by students, especially from Boston University. It has a large proportion of young population as compared to other residential areas.

Back Bay is a residential area that has a large number of young professionals and families. This neighborhood is considered as one of Boston's most expensive residential areas and is also well known for its prominent retail stores.

Fenway is home to many schools and colleges, such as Northeastern University, Simmons College, Emmanuel College, Massachusetts College of Art and Design, Massachusetts College of Pharmacy and Health Sciences and Wentworth Institute of Technology. The area also has a large number of students and some young professionals.

South Boston is a densely populated residential neighborhood in Boston. It is one of the oldest neighborhoods with a large working class population. Figure 2 shows a bird's-eye view of the Back Bay and South Boston area.

The data was collected by war-driving in these neighborhoods using a set of **software / hardware tools**. The hardware consisted of an ASUS Eee PC 1000HE, equipped with three TP-LINK TL-WN722N wireless N150 high gain USB Adapters and a GlobalSat BU-353 USB GPS navigation receiver. The antenna of the adapters was replaced by an Alfa 9dBi WiFi OMNI-directional high-gain antenna for better coverage. We used Kismet to passively collect beacon frames.

Fig. 2: Bird's-eye View of the Back Bay and South Boston.

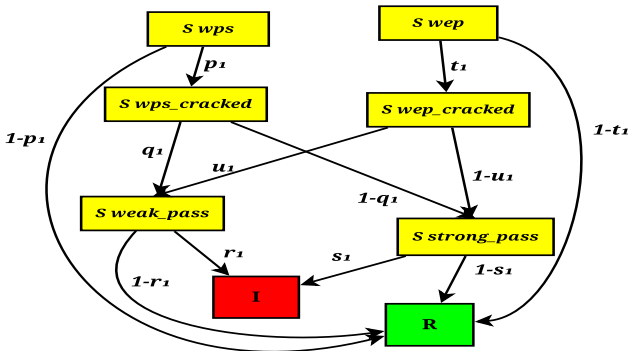


Fig. 3: Diagram of the SIR model we use to investigate the spread of the attack.

Using Kismet with the hardware enabled us to gain substantial data from these neighborhoods even when we were capturing beacon frames in pcap format with PPI GPS header on only channels 1, 6 and 11. The reason we chose these channels was that many APs use these three orthogonal channels to mitigate overlap and interference. To preserve the privacy of users of these wireless networks, we only collected the BSSID, ESSID, signal, channel, latitude, longitude and encryption type.

B. Connectivity Graph

To study and simulate the spread of malware infections over connected WiFi networks, we calculated the connectivity graph of the APs. The location of an AP was chosen as the coordinates where our equipment received the strongest signal from the AP. Two APs, A and B, were connected to each other if they were within R proximity of each other.

The proximity ranges suggested by earlier studies was up to 100 meters [8], [16], but in order to provide a conservative lower bound for the range of connectivity, we developed a tool to estimate the range of APs. The tool computes the convex hull of all the points where each AP was heard. We then computed the diameter of the convex hull. We used this data to calculate the average maximum distance of all APs as an estimate for the radius of connectivity.

C. SIR Spread Model

To analyze the spread of an attack, we developed a SIR compartmental model, Figure 3. The susceptible state of the model was divided, so that we could derive a more realistic and accurate result and get a finer control over the speed of the epidemic. For example, if the susceptible state of the model was not divided, we would have had to consider the time required for cracking WPS and WEP as the same. That would have resulted in inaccurate results. The amount of time it requires to crack WPS is substantially more than the time it requires to crack WEP and we can only achieve accurate results in a divided model.

The model considers various probabilities. For all WPS enabled APs, there is a probability p_1 that a WPS PIN will be successfully cracked in p_{1Stime} minutes and there is a probability $(1 - p_1)$ that a WPS PIN will not be successfully cracked after p_{1Ftime} minutes. If a WPS PIN is not successfully cracked in p_{1Ftime} minutes, we can conclude that the AP is immune / recovered. For all WEP enabled APs, there is a probability t_1 that a WEP key will be successfully cracked in t_{1Stime} minutes and there is a probability $(1 - t_1)$ that a WEP key will not be successfully cracked after t_{1Ftime} minutes. If a WEP key is not successfully cracked after t_{1Ftime} minutes, we can conclude that the AP is immune / recovered. Once a WPS PIN is successfully cracked, there is a probability q_1 that the AP uses a weak administrator password and there is a probability $(1 - q_1)$ that the AP uses a strong administrator password. If a WEP key is successfully cracked, there is a probability u_1 that the AP uses a weak administrator password and there is a probability $(1 - u_1)$ that the AP uses a strong administrator password. If the AP uses a weak administrator password, there is a probability r_1 that the password will be cracked and the AP will be re-flashed / infected in r_{1Stime} minutes and there is a probability $(1 - r_1)$ that the password will not be cracked or the AP will not be re-flashed after r_{1Ftime} minutes. If a password is not cracked or the AP is not re-flashed within r_{1Ftime} minutes, we can conclude that the AP is immune / recovered. If the AP uses a strong administrator password, there is a probability s_1 that the password will be cracked and the AP will be re-flashed / infected in s_{1Stime} minutes and there is a probability $(1 - s_1)$ that the password will not be cracked or the AP will not be re-flashed after s_{1Ftime} minutes. If a password is not cracked or the AP is not re-flashed within s_{1Ftime} minutes, we can conclude that the AP is immune / recovered.

V. Analysis

A. Basic Statistics

A total of 26306 unique AP BSSIDs were collected from **Fenway**. It was surprising to find out that 4093 (15.56%) APs still use WEP even when it is well known to have serious flaws [17], [18], [19]; 3427 APs (13.03%) were open and used no encryption, but that could be because these APs probably belong to institutes that authenticate users later or have guest networks. We identified over 6% of BSSIDs as belonging to institutes in the area. As expected, the majority of APs were WPA / WPA2 protected, even though there are some practical attacks on these protocols [19], [20] as well. WPS

Encryption	Number of APs	Percentage
Allston		
WEP	1667	10.81%
Open	1598	10.36%
WPA/WPA2	12157	78.83%
WPS	6149	50.58%
Back Bay		
WEP	5369	16.38%
Open	5051	15.41%
WPA/WPA2	22367	68.22%
WPS	7809	34.91%
Fenway		
WEP	4093	15.56%
Open	3427	13.03%
WPA/WPA2	18786	71.41%
WPS	5764	30.68%
South Boston		
WEP	1874	12.70%
Open	1110	7.52%
WPA/WPA2	11772	79.78%
WPS	5504	46.76%

TABLE I: Basic statistics of APs, surprisingly a large number of APs still use WEP. WPS is enabled on many of the WPA / WPA2 APs.

was introduced to facilitate users to setup a secure network, therefore to calculate the percentage of WPS enabled devices, it is more reasonable to calculate the percentage of WPA / WPA2 APs that have WPS capability enabled. The number of WPS enabled APs were 5764 which is 30.68% of WPA / WPA2 APs.

A total of 32787 unique BSSIDs were collected from **Back Bay**; 5369 (16.38%) of these APs use WEP, 5051 (15.41%) are open, 22367 (68.22%) use WPA / WPA2 and 7809 (34.91%) APs out of 22367 WPA / WPA2 APs have WPS enabled.

We collected 15422 unique BSSIDs from **Allston**; 1667 (10.81%) of these APs use WEP, 1598 (10.36%) are open, 12157 (78.83%) use WPA / WPA2 and 6149 (50.58%) APs are WPS configured. In **South Boston**, we collected 14756 unique BSSIDs, 1874 (12.70%) use WEP, 1110 (7.52%) are open, 11772 (79.78%) use WPA / WPA2 and 6149 (5504%) APs are WPS enabled. Table I depicts the basic statistics of the APs.

A surprising fact is that all neighborhoods have a similar proportion of WEP, Open, and WPA / WPA2 APs, though the demographics are substantially different. The WPS proportion seems to be slightly different as it is a more recent standard, and it is possible that it can be correlated with the higher household income of the Back Bay area.

B. Connectivity Analysis

To analyze the connectivity, we investigated radii of 15, 30, 50, 75 and 90 meters. 15 meters is common and a normal distance between many users and their APs. Using our radius computation approach based on the convex-hull of measurement data for APs, we estimated 41 meters as the radius of an AP coverage. This radius would significantly increase at off-peak times when there is less interference. For example, based on our experiments at 4 am, a radius of 90 meters becomes much more plausible. At normal transmission power, WiFi signals can easily be heard from a distance of 25 to 50 meters. Most APs are configured to transmit at a lower power than they are capable of. The transmission power can be increased through the administrative interface of the devices. At increased transmission power and at off-peak times, WiFi signals can travel much farther and can be heard from a distance of 100 to 150 meters.

Table II depicts the connectivity statistics of APs using various radii. From the table, we can infer that generally a WEP connectivity graph has more components than a WPS connectivity graph, indicating that the spread of malware with small radii is less practical in WEP networks. However, when we combine WEP and WPS networks, the number of components is reduced to more than half which shows a significant improvement in terms of feasibility of attacks. As expected, with an increase in the radii, the number of components increase in all scenarios (WEP, WPS, WEP+WPS) and the average degree of an AP and the number of edges between the APs increases. An interesting observation is the significant drop in the number of components when we increase the radius from 30 to 40 meters; this means that based on our measurement of lower bound for APs radius, the attacks are practical even during normal hours and the attacks do not have to be triggered during off-peak hours of a day. When we increase the radius to 90 meters, even though we have a single component in WPS, the combined network is better connected.

C. Spread Analysis

We looked at the spread of a malware over wireless networks using WEP and WPS flaws. Based on our measurements and previous studies [16], [8] we considered radii of 50, 75 and 90 meters. According to Reaver, on average it would take about 4-10 hours to recover a WPS PIN, but in practice it takes half this time, therefore we consider $p_1 = 60\%$, $p_{1Stime} = 180min$, $360min$ and $540min$, and $p_{1Ftime} = 10min$. More than 50% of users do not change any of the default configurations [21] (SSID, channel, and credentials) on their APs, therefore we assume q_1 = 50% and $u_1 = 50\%$. We set $t_1 = 100\%$ and $t_{1Stime} = 20min$ [18]. Based on the complexity of the password, we consider r_1 = 10% and $s_1 = 80\%$; based on our experiments, in each second 20-30 admin password can be tried with an AP, therefore we set $r_{1Stime} = t_{1Ftime} = 60min$ and $s_{1Stime} = s_{1Ftime} = 120min$.

Based on the parameters in a single connected component graph, an upper bound of the number of infected APs can be inferred; on an average 32% of the WPS+WEP APs can be infected. The attack has two roots, WPS and WEP. Therefore the total success rate is the sum of the success rate in the two sub-trees.

$$\%WPS \times (p_1 \times q_1 \times r_1 + p_1 \times (1 - q_1) \times s_1) \qquad (1)$$

Back Bay			
Radius	No. of Edges	Avg. Deg.	Conn. Comp.
WEP			
15	111739	42.62	216
30	184821	69.85	32
50	306303	115.10	1
75	533528	199.74	1
90	698785	261.30	1
WPS			
15	152744	40.02	124
30	318474	82.36	20
50	614203	157.90	3
75	1115513	285.97	1
90	1497259	383.49	1
WEP + WPS			
15	425520	65.48	57
30	826137	126.19	11
50	1535754	233.73	1
75	2771477	420.98	1
90	3726970	565.78	1
South Boston			
Radius	No. of Edges	Avg. Deg.	Conn. Comp.
WEP			
15	6874	8.34	437
30	11237	12.99	117
50	17957	20.16	15
75	28599	31.52	2
90	42129	45.96	2
WPS			
15	47917	18.38	277
30	89978	33.64	23
50	150964	55.77	1
75	254583	93.36	1
90	378681	138.38	1
WEP + WPS			
15	83636	23.64	223
30	156472	43.36	10
50	26339	72.43	1
75	443570	121.09	1
90	662227	180.30	1

TABLE II: Connectivity statistics of APs in Back Bay and South Boston, using different radii.

$$\%WEP \times (t_1 \times u_1 \times r_1 + t_1 \times (1 - u_1) \times s_1) \qquad (2)$$

Figure 4 shows the speed and success rate of attacks, based on the defined parameters. As we can see in figure 4 and table II, although we have a single connected component with a radius of 50 meters, the average degree is much lower compared to the higher radii and therefore we cannot expect the infection's spread to reach its theoretical upper limit. As we can see in this scenario, we can infect 19% to 23% of the susceptible APs in 139870 minutes (97.1 days) to 198090 minutes (137.5), respectively. As we increase the radius to 75 meters, we can infect a larger portion of the population; 33% to 35% in 157230 minutes (109.1 days) to 280080 minutes (194.5 days), a trade-off between the infection rate and the amount of time needed.

As we can see, when we increase the radius to 90 meters, we have one highly connected component. In this scenario an attack can get real leverage from the WEP+WPS network, and as the figures show we can infect 34% to 35% in 90080 minutes (62.5 days) to 273500 minutes (189.9 days). Since with an increased radius, the average degree of connectivity and number of connected WEP APs significantly increases, the time needed to crack WPS PIN would not be a dominant constraining factor.

VI. Countermeasures and Related Work

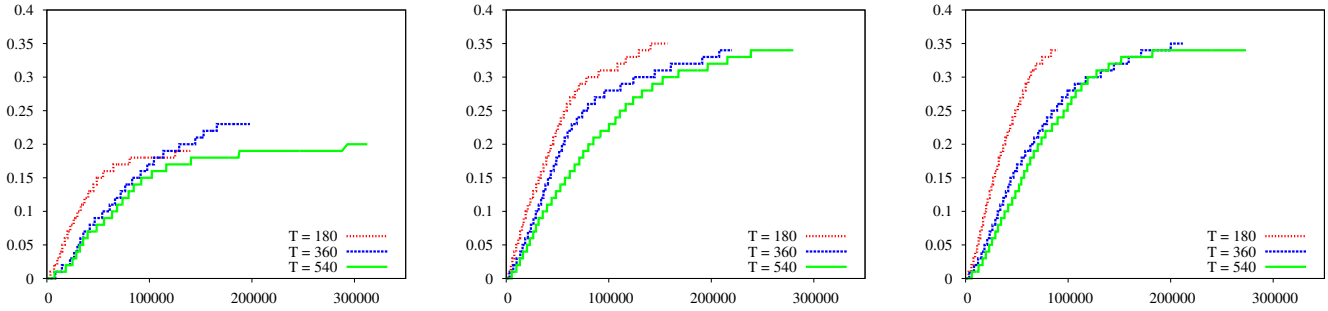
Handheld devices such as smart-phones and tablets use WiFi as their primary mode of communication. Even with high speed 4G connections, mobile users prefer using WiFi for obvious reasons of lower cost, better speed and connectivity. Modern laptop and desktop computers are all equipped with WiFi capability and home networks these days hardly use wired connections. In fact, laptop manufacturers have started manufacturing laptops such as the MacBook Air that primarily rely on WiFi for connectivity. This trend portrays how important it is to promptly provide solutions and countermeasures to address vulnerabilities with WiFi networks.

The simplest solution would be to disable WPS on an AP. Unfortunately, majority of users are not confident in configuring their APs and fear that changes to the configuration may break their AP [22]. What is even more unfortunate is that some APs come with WPS enabled by default with no mechanism to disable it. We investigated different APs, and found out that many new wireless routers come with a pre-configured wireless setup and encourage users to use WPS for connectivity, or the wireless is disabled and users need to configure the AP themselves. APs that come with disabled wireless setup also have wireless security disabled; Since APs still support WEP and it is the first choice in alphabetic order, an unwitting user is more likely to choose this option.

The more complex and sophisticated approach would be to deploy Intrusion Detection Systems that use flow characteristics of WiFi networks to detect anomalies and attack signatures. Tools such as Kismet have features that can detect attack signatures such as a WPS brute-force attack and can be used as an effective Intrusion Detection System to detect such attacks and alert the administrators or users. Beyah et al. [23] investigate the evolution of threats and the possibility of deploying collaborative distributed wireless intrusion detection systems.

Another approach to mitigate the spread of such infections would be to use reliable bootstrap architectures [24], malicious code detectors [25], or automated firmware security upgrades. A new direction of research is heading towards identifying new threat models and malware behavior in WiFi networks.

Channakeshava et al. [26] used activity based models to investigate the spread of Bluetooth worms in a mobile urban population. Based on their investigation, they suggested a framework to generate synthetic data to study the spread of Bluetooth worms over real wireless networks.



(a) R = 50m (b) R = 75m (c) R = 90m

Fig. 4: Graph of spread, considering three different radii (50m, 75m, and 90m), with 3 different duration of time to recover WPS PIN.

Hu et al. [8] used war-driving data collected by the Wireless Geographic Logging Engine (WiGLE) to investigate the spread of malware over WiFi routers in urban areas of United States by exploiting the design flaws in WEP.

Tsow et al. [11] introduced the notion of warkitting, a method of subverting wireless APs using unauthorized mobile WiFi clients. They evaluated the feasibility of implementing such an attack and demonstrated how such attacks can increase the volume of credential theft, even more than phishing.

Cui and Stolfo [27] provide a quantitative lower bound on the number of vulnerable embedded devices. They identified and investigated over 540,000 (over 13% of all identified embedded devices) publicly accessible devices across 144 countries, including firewalls, routers, VoIP adapter etc. that used factory default root passwords.

Akritidis et al. [16] studied various attacks on wireless networks over populated metropolitan areas and tried to quantify the threats from these attacks. They investigated three scenarios namely "Wildfire" virus spread over wireless networks, citywide phishing attacks and rouge APs that compromise location privacy. Their study was based on real world data from USA and Singapore. They demonstrated that 80% of wireless devices can be infected in 20 minutes and that an adversary can track the location of wireless users with only 1000 zombies.

Gu et al. [28] studied the propagation of worms over hidden wireless networks. They focused on the potential wireless connection topology that could assist the spread of a worm in dense and overlapping wireless networks. They proposed a model to investigate the spread and speed of such worms. They also used real data to simulate the spread of these worms and validate their models.

Sun et al. [29] studied the propagation of worms in WSNs and presented a baseline worm model in this context. They looked at the impact of various network protocols and configurations on the model of spread, using simulation to characterize a suitable model.

VII. Conclusion and Future Work

In this work, we looked at the design flaws in WPS and how it can give an adversary leverage to compromise a connected network of APs. We passively collected beacon frames from APs in four neighborhoods of Boston, and we developed a SIR model to study the spread of an infection that exploits WPS and WEP vulnerabilities. Our experimental study showed the feasibility of such attacks, especially when WPS flaws are supplemented with WEP vulnerabilities. Interestingly, all the neighborhoods explored exhibit very similar infection and spreading characteristics even when they have distinct population demographics.

WEP is known to be flawed and well-studied, hence WPA was developed by Wi-Fi Alliance as a replacement; to facilitate users to setup a secure network. Wi-Fi Alliance introduced WPS in 2007, but it was recently discovered to be flawed and this flaw allows an adversary to bypass WPA / WPA2 security measures. As previous studies have shown [22], many users are not comfortable with configuring their APs in fear of breaking it. Many non tech savvy users are familiar with flaws in WEP, and prefer to simply use WPA instead, but unfortunately even many tech savvy users are not aware of the WPS flaw. Another problem with WPS is that it is enabled by default and on some APs it is not possible to disable this capability, which makes the problem more critical. More intuitive authentication mechanisms should be used that would facilitate the setup of secure connections, Cassola et al. [30] suggest such new approaches.

For a long time APs have been considered as a self contained box. Current advancement in field networking advocates separation of control plane and data plane. Kim and Feamster [31] investigate three problems in network management: support for higher level language configuration, better visibility and frequent changes to network state. They designed and implemented a framework to facilitate network management based on Software Defined Networking (SDN). Jafarian et al. [32] investigated deployment of SDN to mitigate network discovery and scanning by an adversary. Recent studies such as Odin [33] designed and implemented WPA2 Enterprise support on APs using SDN principles.

Many new wireless enabled systems are leaving the traditional designs and use the mechanism suggested above. For example, Meraki produces many network appliances, including wireless APs that have central cloud based control, which provides flexible administration. Roku produces digital media receivers that are updated regularly from the cloud by the producer. Although these approaches provide more flexibility and better opportunity, they are still mostly closed to third-parties and do not provide interoperability; Yet, incorporating a safe-update component to allow manufacturers and ISPs to perform critical security to APs in a seamless manner, like Roku, could significantly help to mitigate many such attacks that exploit misconfiguration and policies.

Based on this study and previous works by researchers, we suggest that traditional view of APs should change. We advise the separation of data plane and control plane in APs as well. Today many APs provide connectivity to more devices than before and many households use more than one AP, or could greatly benefit from use of more than one. Separation of planes at APs, can make them more flexible, powerful and easier to maintain, namely: easier firmware upgrades, better handoff and mobility support between multiple APs, possibility of connection sharing, new authentication mechanisms and more flexible QoS support. We advocate new design and framework, that enables standard, open and scalable configuration and administration.

REFERENCES

[1] M. Ramsay, "Wi-Fi offload rising amid soaring data traffic," 2012, http://www.wirelessweek.com/News/2012/07/technology-WiFi-Offload-Rising-Amid-Soaring-Data-Traffic/.

[2] C. Heffner, "Remote attacks against soho routers," 2010. [Online]. Available: https://www.defcon.org/images/defcon-18/dc-18-presentations/Heffner/DEFCON-18-Heffner-Routers-WP.pdf

[3] S. Viehbock, "Wi-Fi protected setup pin brute force vulnerability," CERT Vulnerability Note VU#723755, December 2011. [Online]. Available: http://www.kb.cert.org/vuls/id/723755

[4] "Gaining a foothold by exploiting VxWorks vulns," Dark Reading, August. [Online]. Available: http://www.darkreading.com/blog/227700867/gaining-a-foothold-by-exploiting-vxworks-vulns.html

[5] W. H. Murray, "The application of epidemiology to computer viruses," *Comput. Secur.*, Apr. 1988.

[6] J. D. Murray, *Mathematical Biology : I . An Introduction , Third Edition*. Springer, 2002.

[7] Y. Kuznetsov and C. Piccardi, "Bifurcation analysis of periodic seir and sir epidemic models," *Journal of Mathematical Biology*, vol. 32.

[8] H. Hu, S. Myers, V. Colizza, and A. Vespignani, "Wifi networks and malware epidemiology," *Proceedings of the National Academy of Sciences*, 2009.

[9] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Phys. Rev. Lett.*, Apr 2001.

[10] "Open infrastructure." [Online]. Available: http://www.ccs.neu.edu/home/noubir/projects/openinfrastructure/

[11] R. A. Tsow, A. Tsow, M. Jakobsson, L. Yang, and S. Wetzel, "Warkitting: the drive-by subversion of wireless home."

[12] M. C. Ang Cui and S. J. Stolfo, "When firmware modifications attacks: A case study of embedded exploitation," in *In Proceedings of the Network and Distributed System Security Symposium, NDSS '13*.

[13] B. Dupasquier, S. Burschka, K. McLaughlin, and S. Sezer, "Analysis of information leakage from encrypted skype conversations," *Int. J. Inf. Secur.*, vol. 9, no. 5, Oct. 2010.

[14] X. Wang, S. Chen, and S. Jajodia, "Tracking anonymous peer-to-peer voip calls on the internet," in *Proceedings of the 12th ACM conference on Computer and communications security, CCS '05*.

[15] N. Husted and S. Myers, "Mobile location tracking in metro areas: malnets and others," in *Proceedings of the 17th ACM conference on Computer and communications security, CCS '10*.

[16] P. Akritidis, W. Y. Chin, V. T. Lam, S. Sidiroglou, and K. G. Anagnostakis, "Proximity breeds danger: Emerging threats in metro-area wireless networks," in *In Proceedings of the 16 th USENIX Security Symposium, SS '07*.

[17] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," in *Proceedings of the 7th annual international conference on Mobile computing and networking, MobiCom '01*.

[18] A. Bittau, M. Handley, and J. Lackey, "The final nail in wep's coffin," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy, SP '06*.

[19] E. Tews and M. Beck, "Practical attacks against wep and wpa," in *Proceedings of the second ACM conference on Wireless network security*, ser. WiSec '09.

[20] A. Cassola, W. Robertson, E. Kirda, and G. Noubir, "A practical, targeted, and stealthy attack against wpa enterprise authentication," in *In Proceedings of the Network and Distributed System Security Symposium, NDSS '13*.

[21] R. Shah and C. Sandvig, "Software defaults as de facto regulation: The case of wireless aps," in *In The 33rd Research Conference on Communication, Information, and Internet Policy*, 2005.

[22] A. L. Stephano and D. P. Groth, "Useable security: interface design strategies for improving security," in *Proceedings of the 3rd international workshop on Visualization for computer security*, 2006.

[23] R. Beyah, C. Corbett, and J. Copeland, "The case for collaborative distributed wireless intrusion detection systems," in *IEEE International Conference on Granular Computing, GrC '06*.

[24] W. A. Arbaugh, D. J. Farber, and J. M. Smith, "A secure and reliable bootstrap architecture," in *Proceedings of the 1997 IEEE Symposium on Security and Privacy, SP '97*.

[25] F. Adelstein, M. Stillerman, and D. Kozen, "Malicious code detection for open firmware," in *Proceedings of the 18th Annual Computer Security Applications Conference, ACSAC '02*.

[26] K. Channakeshava, D. Chafekar, K. Bisset, V. S. A. Kumar, and M. Marathe, "Epinet: a simulation framework to study the spread of malware in wireless networks," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Simutools '09*.

[27] A. Cui and S. J. Stolfo, "A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan," in *Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10*.

[28] B. Gu, X. Hong, and P. Wang, "Modeling worm propagation through hidden wireless connections," in *Proceedings of the 28th IEEE conference on Global telecommunications, GLOBECOM'09*.

[29] B. Sun, G. Yan, and Y. Xiao, "Worm propagation dynamics in wireless sensor networks," in *IEEE International Conference on Communications, ICC '08.*

[30] A. Cassola, T. Jin, H. Kumar, G. Noubir, and K. Sharma, "Demo: Sneap: a social network-enabled eap method no more open hotspots," in *Proceedings of the 9th international conference on Mobile systems, applications, and services, MobiSys '11*.

[31] H. Kim and N. Feamster, "Improving network management with software defined networking," *Communications Magazine, IEEE*, 2013.

[32] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: transparent moving target defense using software defined networking," in *Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12*.

[33] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise wlans with odin," in *Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12*.