2014 IEEE Conference on Communications and Network Security

CETAD: Detecting Evil Twin Access Point Attacks in Wireless Hotspots

Hossen Mustafa
University of South Carolina
Email: mustafah@cse.sc.edu

Wenyuan Xu *
University of South Carolina, Zhejiang University
Email: wyxu@cse.sc.edu

***Abstract*—Wireless hotspots allow users to use Internet via Wi-Fi interface, and many shops, cafés, parks, and airports provide free wireless hotspot services to attract customers. However, there is no authentication mechanism of Wi-Fi access points (APs) available in such hotspots, which makes them vulnerable to evil twin AP attacks. Such attacks are harmful because they allow to steal sensitive data from users. Today, there is no client-side mechanism that can effectively detect an evil twin AP attack without additional infrastructure supports. In this paper, we propose a mechanism *CETAD* leveraging public servers to detect such attacks. CETAD only requires installing an app at the client device and does not require to change the hotspot APs. CETAD explores the similarities between the legitimate APs and discrepancies between evil twin APs, and legitimate ones to detect an evil twin AP attack. Through our implementation and evaluation, we show that CETAD can detect evil twin AP attacks in various scenarios effectively.**

Keywords*—*Wi-Fi Security, Wireless Hotspot, Android

I. Introduction

Internet has become a part of our everyday life, and Wi-Fi has gained much popularity in the last few years for accessing Internet. Wi-Fi market reached 6.4 billion in 2011 [1] and a rapid growth is forecasted in the upcoming years [2] as most of mobile devices (e.g., smartphones, tablets, etc.) have Wi-Fi capability. Due to such popularity, many shops, cafés, airports, etc., provide free Wi-Fi hotspot services to attract customers. There are more than $840,000$ Wi-Fi hotspots worldwide [3] and the number is growing everyday. Since the goal of the hotspots is to provide convenience and to attract customers, few or no security mechanism is in place. For instance, in the US, McDonalds, Starbucks, etc., provide free, open, and zero liability[1] Internet access to customers.

The openness of Wi-Fi hotspot makes it vulnerable to evil twin AP attacks [4]. In an evil twin AP attack, the adversary sets up a phishing AP that pretends to be the legitimate one as it uses the same Service Set IDentification (SSID) as a legitimate AP. An evil twin AP attack has been used to launch a Man-in-the-Middle (MITM) attack [5][6] because all the packets from the client to any web server must go through the AP. MITM attacks enable adversaries to hijack a session [6], or steal sensitive data from the user by message falsification [7].

An evil twin AP attack is easy to launch, especially in a Wi-Fi hotspot due to the lack of security mechanisms of hotspots. Evil twin AP attacks can be launched by using a laptop, or a smartphone exploiting the loopholes of Wi-Fi client software implementation: Existing built-in Wi-Fi client implementation assumes that all the APs with the same SSID are legitimate and automatically connects to the AP with the maximum Received Signal Strength Indication (RSSI) value. As a result, if the RSSI of the evil twin AP is higher than that of a legitimate AP, a client will associate with the evil twin AP. Already, there are free apps for smartphone [8] and software for laptop [9] that are available to impersonate a legitimate AP.

Much work addresses the problem of detecting rogue AP[2]. However, most solutions are designed for infrastructure network rather than for client devices. For instance, several mechanisms have been proposed to monitor packets at network gateways or routers [10][11][12][13], or to install extra custom devices for monitoring the network infrastructure, e.g., mobile agents [14], distributed radios [15], trusted wireless clients [16], etc. Those solutions are inapplicable to Wi-Fi hotspots. A Wi-Fi hotspot provides free Internet service to attract customers; it has little motivation to guarantee no attacks nor will setup additional devices or install detection software in their infrastructure to detect an evil twin AP attack. Moreover, most hotspot providers free themselves of any responsibilities from any damages due to security vulnerabilities of their system through “Terms and Conditions”. Thus, *a client side* scheme to detect evil twin AP attacks is practical to ensure security of the Wi-Fi access. To the best of our knowledge, no practical scheme is available that allows a wireless user to check the integrity of an AP at the client side. Existing mechanisms for detecting evil twin AP attacks require to install additional hardware in each hotspot [17] or need prior training [18]. Thus, we focus on designing a plug-and-play mechanism to detect evil twin AP attacks that only requires to install software at the client device.

Designing a client side mechanism to detect evil twin AP attacks is challenging for several reasons. First, the client has limited resources. Neither does it have access nor prior information about the hotspot architecture. Second, hotspots use various Wi-Fi setup and the mechanism must consider all scenarios. Third, adding custom hardware, e.g., routers or servers, is not an option as it would limit the applicability and universal acceptance. We overcome these challenges and design a detection mechanism that we call `CETAD` (Client end Evil Twin Access point Detector). The main idea of CETAD is derived based on the following observation: when multiple APs are legally configured to form a hotspot, they typically

* This author is the corresponding author.

** This work has been funded in part by NSF CNS-0865671.

[1] Customers have to agree to terms and conditions that the provider has no liability in case of security issues during hotspot use.

[2] Evil twin AP is one type of rogue AP. Additionally, rogue AP can be an unauthorized AP connected to a network.

978-1-4799-5890-0/14/$31.00 ©2014 IEEE

connect to the Internet via the same Internet Service Provider (ISP). Thus, besides the same SSID, they also share similar network parameters such as ISP names, Global IP addresses, Round Trip Time (RTT), temporal network behavior, etc. However, when an evil twin AP is present, discrepancies can be found among the APs because the evil twin AP uses a different network setup. Our detection mechanism explores such similarities and discrepancies to verify whether all available APs belong to the same group. In particular, we use three statistics in our detection mechanism: similarity of ISP information, difference in RTT values, and standard deviation of RTT values. Using these measurements, we detect an active evil twin AP attack by identifying multiple groups of APs with one group containing the legitimate AP(s) and the other group(s) containing the evil twin AP(s). Even though CETAD is designed for client devices, it can be extended to detect evil twin APs in an infrastructure network as well.

II. Background

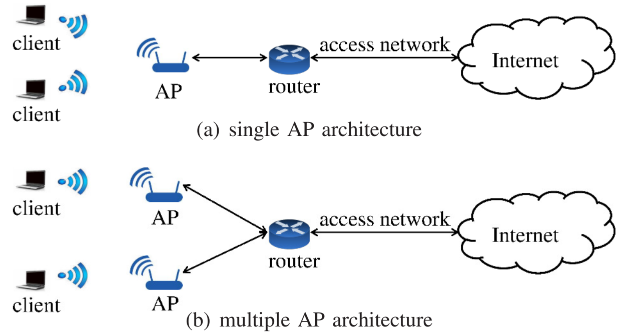
We consider Wi-Fi hotspots only because hotspots generally allow open access and are vulnerable to evil twin AP attacks. Since our detection scheme leverages network elements and characteristics, we give a brief overview of Wi-Fi hotspot architecture below.

IEEE 802.11 standards [19] (popularly known as Wi-Fi) defines the communication protocol between a client and an AP. To access Internet, a client first associates with an AP by selecting the desired SSID. If multiple APs have the same SSID, the client associates with the AP that has the highest RSSI. Once associated, the client gets network parameters for the hotspot using Dynamic Host Configuration Protocol (DHCP) [20](Section II-B) and then, can start surfing Internet.

The popularity of Wi-Fi has made it one of the must-have properties for laptops, tablets, etc., for accessing Internet. A typical Wi-Fi network has four types of components: an access network, a router, a Wi-Fi AP, and one or more wireless clients, as shown in Figure 1. Many times APs are mistakenly considered as a router that connects an access network. In fact, an AP works in Open Systems Interconnect (OSI) layer 1-2, and connects wireless clients to a Local Area Network (LAN). Then, a router, a layer 3 device with Network Address Translation (NAT) capability, serves multiple LAN clients and connects to the Internet through one of the following access networks: (a) cable networks, (b) digital subscriber line (DSL) networks, (c) mobile networks, (d) Ethernet, and (e) optical fiber networks. Among these access networks, cable network has the largest market share in hotspots in the US. CableWiFi initiative[3] has $165,000$ Wi-Fi hotspots that uses cable networks in the US [21]. AT&T has more than $32,000$ Wi-Fi hotspots in the US that mostly uses DSL and cable networks [22]. Regardless of which access network is used, a wireless hotspot adopts one of two network architectures and supports DHCP, as we discuss in the following.

A. Hotspot Architecture

1) Single AP Architecture: In this setup, there is one AP that supports multiple wireless clients. As shown in Figure 1(a), the AP is connected to a modem or a router through a Wide Area Network (WAN) interface. The modem/router is connected to the ISP via one of the possible access networks.



(a) single AP architecture

(b) multiple AP architecture

Figure 1. A simple Wi-Fi network in single, and multiple AP architecture where the APs are connected to Internet via an access network.

2) Multiple AP Architecture: The range of an AP is limited. For instance, an 802.11b AP has range of 120 ft indoors and 300 ft outdoors [19]. As a result, to cover a large area, multiple APs are required. The APs form a set named `Extended Service Set (ESS)` and all the APs in the ESS have the same SSID. Additionally, all the APs generally have similar configurations so that a user can automatically switch to another AP with a higher RSSI value when he moves towards the new AP. The APs are not required to have routing capabilities, instead they are connected to a router that is connected to the ISP via an access network. A simple Wi-Fi network for this architecture with two APs is shown in Figure 1(b). Alternatively, the APs can be configured in wireless distribution system (WDS) [19] mode. In WDS, only one AP is connected to the router and all APs can communicate among themselves over wireless channels. However, WDS has not been standardized yet and most WDS capable APs do not support inter-operability. Thus, in this paper, we do not consider multiple AP architecture with WDS mode.

3) Summary: A hotspot can implement either single or multiple AP architecture based on the area of coverage. However, there is no way for a wireless client to determine the architecture of a hotspot without administrative access. Thus, evil twin AP attacks are possible in both architectures. For multiple AP architecture, most Wi-Fi client software show only the AP that has the highest RSSI value and allow the users to associate with it without validation.

B. Automatic Network Configuration

A Wi-Fi client requires to configure network interface for using Internet and most Wi-Fi networks have at least one DHCP server for automatic network configuration. The DHCP protocol works as follows. The client broadcasts a `DHCP DISCOVER` packet and each DHCP server in the Wi-Fi network responds with a `DHCP OFFER` to the client. The client accepts one offer by broadcasting a `DHCP ACK`. A DHCP offer contains at least 5 fields: (a) Host IP address, (b) Subnet mask, (c) Router IP address, (d) DNS [23] servers, and (e) Server identifier. The client uses these information to configure his Wi-Fi interface automatically and then, starts surfing Internet. Our method utilizes DHCP to automatically configure the network interface.

[3] The partner companies are Cablevision Systems, Comcast, Time Warner Cable, Cox Communications and Bright House Networks.

III. Evil Twin AP Attacks

In a Wi-Fi network, the AP periodically broadcasts SSID, which allows a Wi-Fi client to discover the existence of the AP and associate with it. In an evil twin AP attack, the adversary sets up her AP using the SSID of the targeted Wi-Fi network. As a result, a client receives SSID broadcast from both the legitimate AP and the evil twin AP, but it cannot differentiate between these APs. The client simply assumes that both the APs are legitimate and associates with the one that has a higher RSSI value. For a successful attack, an adversary can increase the transmission power of the evil twin AP, or set it closer to the client to ensure that the client gets a higher RSSI value for the evil twin AP. There are several ways an adversary can launch an evil twin AP attack.

A. Launching Evil Twin AP Attacks

1) Attacks using Mobile Internet Access: For this attack, the adversary uses mobile Internet, e.g., 2G/3G/4G, as the access network for connecting to the Internet. We denote these attacks as `Mobi Attacks` and depict this model in Figure 2(a). There are two types of devices an adversary can use to launch Mobi attacks:

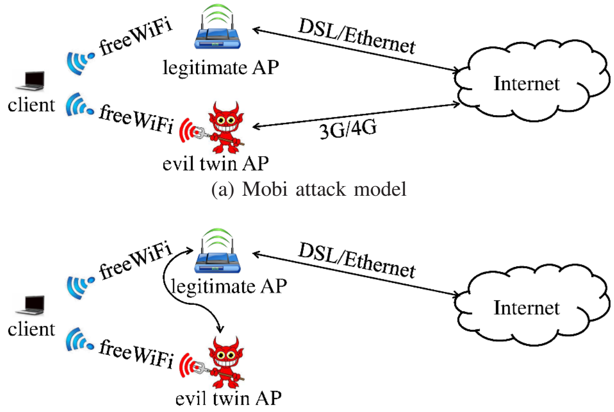
Hotspot Router. The adversary can use an off-the-shelf device to create an evil twin AP, e.g., Linksys WRT54G-TM hotspot router. The adversary can simply configure the SSID of the AP to one of the Wi-Fi hotspots.

Smartphone. A smartphone can act as an AP and thus can allow Wi-Fi clients to use mobile Internet service of the smartphone. Sharing mobile Internet connection via Wi-Fi interface is also known as tethering [8]. The adversary can configure a tethering app with the SSID of the hotspot and turn it on, e.g., Android Wi-Fi Tether app.

2) Attacks utilizing the Victim AP's Internet Access: This attack eliminates the requirement of a device with mobile Internet service. In this attack, the adversary associates with a victim AP as a Wi-Fi client and shares this Internet connection as an AP. We denote these attacks as `Multihop Attacks`, as shown in Figure 2(b). An adversary can launch this attack in the following two ways.

Single Wi-Fi Interface (`Si-Fi Attack`). The adversary can install a software to use a laptop as an AP with Wi-Fi hotspot SSID, e.g., `Virtual Router` [9]. This software creates two virtual Wi-Fi interfaces, one for associating with the legitimate AP and the other for acting as an evil twin AP, using a single physical interface. In this case, the evil twin AP must use the same wireless channel as the legitimate AP.

Dual Wi-Fi Interfaces (`Du-Fi Attack`). In this model, the adversary has two Wi-Fi interfaces. For instance, she can use the built-in Wi-Fi interface and use an additional USB Wi-Fi interface. The adversary can configure one Wi-Fi interface in ad hoc mode using the SSID of the Wi-Fi hotspot. In the ad hoc mode, the device can communicate with other Wi-Fi devices directly and thus pose as an AP. The adversary can use the other Wi-Fi interface to associate with the legitimate AP and share its Internet connection. In this case, the evil twin AP can use any wireless channel.



(a) Mobi attack model

(b) Multihop attack model

Figure 2. In Mobi attack scenario, the adversary uses a Wi-Fi interface to create an evil twin AP which connects to Internet through 3G/4G. In multihop attack scenario, she creates an evil twin AP and connects to Internet through the legitimate AP.

B. The Consequence of Evil Twin AP Attacks

An evil twin AP attack can be used for stealing sensitive data from a user. For instance, session hijacking can be used to hack an email account, facebook account, etc [24]. This attack is feasible because most users enable *automatic login* to their online accounts (e.g., email accounts) to avoid repeated password entry. In such a configuration, a session key is saved both on the client side and server side for each account. To steal the session key, the adversary simply injects `<img>` element in the HTTP response message for a user. For instance, let the victim has automatic login setup for `mail.yahoo.com` and he is associated with an evil twin AP. When the victim visits a webpage, e.g., `www.bing.com`, the adversary will inject a `<img src="http://mail.yahoo.com">` element in the HTTP response message. As a result, the victim's browser will transmit yahoo cookies. Thus, the adversary will be able to sniff the cookies and use these to log in to victim's `mail.yahoo.com` account. MITM attacks are possible even when the victim uses Wi-Fi Protected Access (WPA) [6]. For instance, the adversary can be an insider and have the WPA key, e.g., hotel customer.

C. Threat Model

It is easy for an adversary to create an evil twin AP in a Wi-Fi hotspot using a Wi-Fi-enabled device, e.g., laptop, smartphone, etc. The adversary may have her own access network, e.g., mobile Internet, for Internet connectivity or she may connect her device to a legitimate AP for accessing the Internet. We envision that the evil twin AP may have the following capabilities to avoid detection: it can (a) control the timing for sending out a packet from the client, (b) reply the client with locally generated spoofed packet, (c) modify packet content while forwarding a packet to the client, or (d) use Virtual Private Network (VPN) for tunneling. Irrespective of the above capabilities, the adversary should forward the packets from an associated client for successful attacks. We assume that hotspot providers are trustworthy as they are bounded by several laws [25] and do not collude with the adversary in any way.

IV. CETAD Overview

We aim at detecting evil twin AP attacks in wireless hotspots from a client end without any infrastructure support. In this section, we identify design requirements, specify hotspot features, give an overview of CETAD, and discuss our observations from several hotspots.

A. Design Requirements

A client-side mechanism for detecting evil twin AP attacks must fulfill the following requirements.

1) The mechanism must not require any administrative access to the routers or APs. The client may have such administrative control in his home but not in a hotspot.
2) It must be able to verify an AP in a hotspot and thus cannot assume any custom infrastructure support, e.g., install extra wireless devices or wireless sensors. Designing a solution with infrastructure support would require hotspot owners to modify hotspots, which is unlikely to happen because most hotspots are free services.
3) It must not use any sensors or interfaces available only in smartphones so that it can be integrated in any type of Wi-Fi enabled devices. Using a resource that is not universal to all Wi-Fi devices will limit its use.
4) It must be mostly automated with little intervention from users to ensure usability.

B. Hotspot Features

We observed that many hotspots share the following features (details in Section IV-D):

(a) The wireless hotspot supports a DHCP server that dynamically assigns network parameters to the clients, e.g., IP address, DNS address, etc.
(b) The wireless hotspot does not use WDS architecture.
(c) The adversary can associate with a legitimate AP similar to a client, but cannot gain administrative access to the infrastructure of the hotspot.
(d) The hotspot uses one ISP for Internet connectivity.

C. CETAD Framework Overview

CETAD is designed based on the idea that the ISP, public IP address, RTT values of packets travelling through two legitimate APs are similar (i.e., the same ISP), but these are different for a legitimate AP and an evil twin AP. This holds for the following cases:

i. *No Attack.* All the legitimate APs will have same ISP and IP address because an organization generally purchase Internet service from one local ISP. Additionally, since the APs will share the same access network, they will have similar RTT values.
ii. *Mobi Attacks.* An adversary uses her own access network (i.e., mobile Internet) for these attacks. Since legitimate APs rarely use mobile Internet due to its cost, the global IP address, and ISP of the legitimate AP and the evil twin AP will be different. Additionally, since the access networks of the legitimate AP and the evil twin AP are different, the RTT values are likely to be different as well.
iii. *Multihop Attacks.* In these attacks, an adversary utilizes the access network of the wireless hotspot; thus the ISP information will not reveal the attack. However, there will still be dissimilarity between a legitimate AP and an evil twin AP in RTT values as the evil twin AP will use a legitimate AP as the next hop.

While the idea is simple, there are several challenges to be addressed: In what degree the IP address and ISP information differ among APs? Can we measure RTT without custom servers? How effectively can RTT be used for detecting an attack? Can an adversary manipulate packets to avoid detection? We address these challenges in CETAD and answer the questions in Section IV-D.

As outlined in Algorithm 1, CETAD works in two phases: (a) secure data collection phase, and (b) detection phase.

Secure Data Collection Phase. In this phase, CETAD communicates with public servers leveraging widely used Hypertext Transfer Protocol Secure (HTTPS) protocol (details in Section V). First it scans the wireless hotspot to detect available APs with the desired SSID. From the AP list, it eliminates APs with low RSSI value by applying a threshold δ. Then, for each of the available APs with RSSI value greater than δ, the mechanism collects data in the following 3 steps.

Step 0: It associates with an AP and collects DHCP information of the Wi-Fi network. This is the first step because for collecting ISP and RTT data, the client must connect to Internet by associating with an AP.

Step 1: It collects ISP information of the AP by contacting a public server. This information is used in ISP-based scheme to find correlation in ISP information of the APs.

Step 2: It collects multiple RTT values by creating an HTTPS connection to a public server. The RTT values are used in the timing-based scheme.

Detection Phase. CETAD detects evil twin AP attacks by classifying multiple APs in a hotspot. After the data collection phase completes, CETAD uses a multi-level approach to classify the APs based on data of all APs. First, it applies ISP-based scheme which utilizes ISP data to identify whether two APs are using the same ISP for Internet access and this scheme is enough to detect *Mobi attacks* (Section III). In the next level, CETAD uses a timing-based scheme utilizing the RTT values to detect both *Si-Fi attacks* and *Du-Fi attacks* (Section III). We discuss the details of ISP-based and timing-based detection schemes in Section V.

Algorithm 1 Evil Twin AP Attack Detection

```
function CETAD($\delta$)                                  ▷ $\delta = rssi\_threshold$
    $ap.list = Scan()$
    for each AP $a$ in $ap.list$ do
        if $a.rssi \geq \delta$ then
            $data[a] = CollectData(a)$
        end if
    end for
    $detect = PerformISPDetection(data)$
    if $detect == false$ then
        $detect = PerformTimingDetection(data)$
    end if
    return $detect$
end function
```

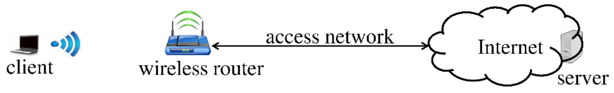
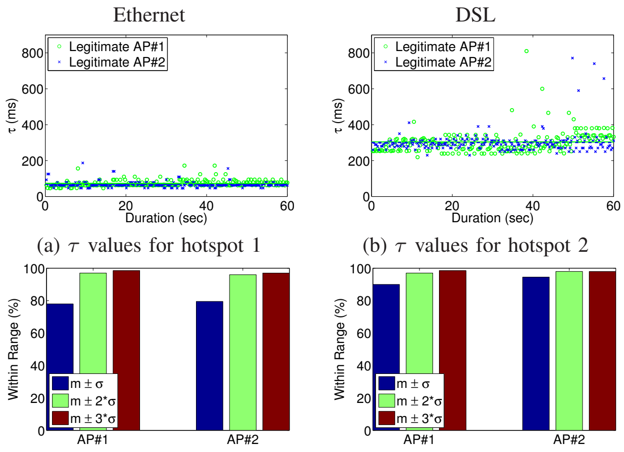


Figure 3. Experimental setup for data connection.



(a) τ values for hotspot 1 (b) τ values for hotspot 2

(c) τ distribution for hotspot 1 (d) τ distribution for hotspot 2

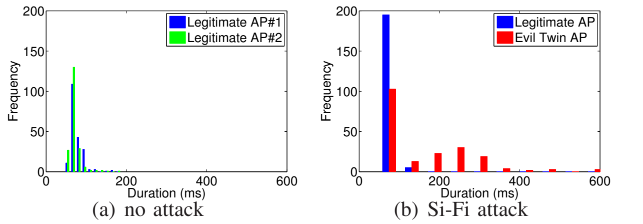
Figure 4. Timing data analysis of 2 legitimate APs in 2 different hotspots in no attack scenario: hotspot 1 uses Ethernet and hotspot 2 uses DSL as access network. Each hotspot consists of 2 APs.

D. Observations

To understand hotspot features and validate our intuition, we studied 30 hotspots which includes Mcdonalds, Wendys, Starbucks, universities, airport, etc. For each hotspot we collected statistics that are used to differentiate APs, and they include ISP information, e.g., public IP address, zip code, ISP name, etc., and timing data for public servers (Section V-A). The use of public server allows our design to be independent of any custom server. Since it is hard to measure packet level RTT using public server, we redefine RTT as the time to setup an HTTPS connection to a public server and denote it as τ. We collect data in following scenarios: no attack, Si-Fi attacks, and Du-Fi attacks (Section III-A2) and show the experimental setup in Figure 3. We collected data from several locations in the USA and in China. We ran the experiments on both weekdays and weekends, and collected approximately 2000 hours of data. In total we collected more than $5,000,000$ instances of data for the five types of access network. We summarize our observations in the following.

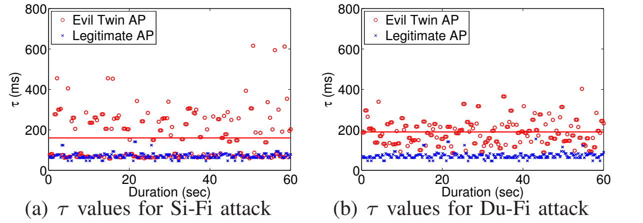
1) Network Analysis: Based on the data of all hotspots we observe that (a) most of the hotspots have two or more APs, (b) each hotspot is connected to one ISP and supports DHCP, (c) all the APs in a hotspot use similar configurations, e.g., shared SSID, global IP address, DNS, etc., which is a standard practice to ensure efficient service [19][26], i.e., to allow users to associate with another AP smoothly when he changes his location, (d) the ISP information of a legitimate AP, and an evil twin AP is different, (e) none of the hotspots utilizes a WDS architecture or uses mobile Internet (3G/4G), (f) the built-in Wi-Fi drivers of all our devices associate with the AP that has the highest RSSI value.

These observations validate our intuition and indicate that ISP information can be useful for attack detection.



(a) no attack (b) Si-Fi attack

Figure 5. Histogram analysis of τ value for no attack and Si-Fi attack scenarios where the hotspot uses Ethernet as access network.



(a) τ values for Si-Fi attack (b) τ values for Du-Fi attack

Figure 6. Timing data analysis for Si-Fi and Du-Fi attack scenarios where the hotspot uses Ethernet as access network.

2) Timing Analysis: From the timing data, we observe that τ values are different for different types of access network in regular (no attack) and attack scenarios.

No attack scenario. We depict τ values for two different hotspots in Figure 4(a)-(b), where one hotspot uses Ethernet and the other uses DSL as the access network. As we can see, the average of τ for one-minute period is approximately $70ms$ for Ethernet and $300ms$ for DSL network. Here, τ values vary over time due to network load, number of users, etc., but the variance is small within a short time interval. Additionally, τ values for different APs in the same hotspot are similar and the mean values are close. As we see in Figure 5(a), τ values for both the legitimate APs are centered around $70ms$ mark. Even though there are some random spikes in τ values, majority of the values can be grouped within a range using mean (m) and standard deviation (σ) of the values. For instance, in both legitimate scenarios (two hotspots), more than 98% of the values are within a range of $m \pm 2 * \sigma$. We depict the percentage of τ values within range for different ranges in Figure 4(c)-(d).

Attack scenario. In Si-Fi and Du-Fi attack scenarios, the τ values through an evil twin AP varies widely compared to a legitimate AP as depicted in Figure 6. Additionally, the mean of τ is much larger for an evil twin AP than for a legitimate AP for both attack scenarios. The histogram in Figure 5(b) shows that τ values are distributed over a long range with high variance for an evil twin AP in Si-Fi attack scenario. We observe similar distribution in Du-Fi attack scenario as well. Several factors may cause such high variance of τ values in the attack scenarios. In Si-Fi attacks, there are additional software processing delay. Additionally, since two virtual interfaces use the only physical interface and single wireless channel for communication, there could be additional delay due to added collisions in the Medium Access Control (MAC) [19] layer. In Du-Fi attacks, the increase in delay could be caused by packet forwarding delay from one interface to another.

The above observation suggests that it is feasible to use τ values effectively for detecting evil twin AP attacks.

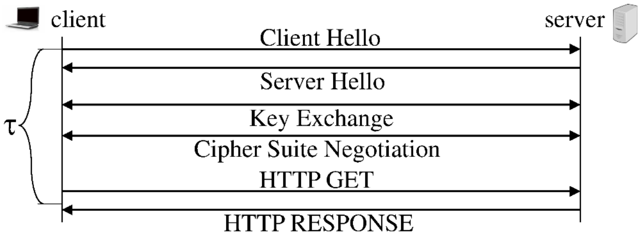


Figure 7. Simplified HTTPS connection setup protocol between the client and the server. τ is the HTTPS connection setup time between the client and the server.

V. CETAD Description

In this section, we discuss the secure data collection phase, and detection phase of CETAD and perform security analysis.

A. Secure Data Collection

In this phase, CETAD collects data for all available APs in the hotspot that have the same SSID. After associating with an AP in a hotspot, CETAD leverages public HTTPS servers to find ISP information of the AP and to measure τ. HTTPS allows us to ensure the integrity of the packets sent and received by CETAD, since an evil twin AP can try to change or manipulate the content of the packets to avoid detection. In the following, we first give an overview of HTTPS and then discuss data collection steps.

1) HTTPS: HTTPS is a secure communication protocol which is designed by adding the security capabilities of SSL/TLS[4] protocol [27] over HTTP. HTTPS is an application layer protocol and uses public key infrastructure (PKI) to exchange short term session keys between the client and the server. Session keys are used to encrypt the data packets that are transmitted over HTTP. A simplified HTTPS connection setup process is depicted in Figure 7. First, the client sends a `Hello` message to the server and the server replies with a `Hello` message. Then the client and the server exchange their public keys upon validation, and agree on cipher suites to use for communication. In these steps, the server creates a unique hash and encrypts it using both the client's public key and its private key, and sends this back to the client. This ensures that only the client is able to read the hashed data. At this point, the connection setup is complete, and the client and the server can securely exchange information.

2) Collecting Data: ISP information for an IP address is publicly available through several HTTPS servers and CETAD uses HTTPS GET to collect the data for the AP. To calculate τ, CETAD measures the difference between initiating an HTTPS connection and establishing the connection. In particular, CETAD measure the difference between the moments when the `Client HELLO` is sent and the one when `HTTP RESPONSE` is received. We choose this difference as τ because it ensures that secure connection is established properly. Additionally, this will also simplify implementation process as most HTTPS library APIs hide the key negotiation from the user. CETAD records n values of τ for each AP.

B. Detection Phase

1) ISP-based Detection.: In order to provide Internet service, a wireless hotspot must have a gateway with a global IP address. Each ISP gets a block of IP addresses and provides at least one unique global IP address to the customers upon subscription. Internet Assigned Numbers Authority (IANA) [28] is the authority to manage global IP addresses all over the world. The information related to each global IP address is public, e.g., the name of the organization it is assigned to, assignment date, location, etc. These information are publicly available in various website, e.g., `www.arin.net`, `www.ip2location.com`. CETAD sends an HTTP GET request to such a server and from the HTTP REPLY packet, it gets several information including the source IP address of that AP's network, ISP information of the source IP address, location, etc. We show some public information obtained by this method for two random IP addresses in Table I.

In wireless hotspots, the legitimate APs are connected to the same router sharing the same global IP address or at least the same ISP. CETAD uses these public IP information to differentiate between APs, i.e., to detect whether two APs use the same IP address block from a ISP. Using this scheme, CETAD can easily detect an evil twin AP attack when the adversary launches a Mobi attack. CETAD matches one or more information from Table I to classify the APs and thus detects an evil twin AP attack.

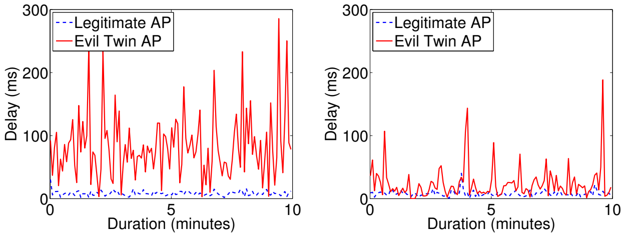
2) Timing-based Detection.: In this section, we discuss how we can use τ values for evil twin AP attack detection. As we discussed in Section IV-D, in non attack scenarios, the values of τ for the same hotspot are similar within a short interval. On the other hand, when an evil twin attack is launched utilizing the victim's Internet (Section III-A2), τ varies significantly. We utilize the temporal values of τ to detect an evil twin attack in two ways: one is by using unsupervised clustering to cluster data in more than one group, and the other is by investigating the standard deviation of the data. We discuss these approaches in the following.

Unsupervized Clustering. In this scheme, we try to cluster APs in a hotspot based on the similarity in the τ values. Since the τ values can vary significantly based on time, hotspot location, network load, etc., a naive threshold-based approach would be ineffective. So, we employ *unsupervised clustering* algorithm for effective detection of evil twin AP attacks. In particular, we use Mean Shift Clustering (MSC) algorithm [29] for clustering the APs which does not require any prior training. We observed some random spikes in τ values (Figure 4) for different hotspots. Since such noisy data can affect the clustering algorithm, in our scheme, we first remove noisy τ values for each AP by applying a filter of $m \pm k * \sigma$, where m is the mean, and σ is the standard deviation

Table I. Sample public information for global IP addresses.

Info	208.54.44.158	171.65.95.202
Location	Doraville, GA	Palo Alto, CA
ZIP Code	30340	94301
Net Speed	DSL	COMP
Usage Type	MOB	EDU
Domain	t-mobile.com	stanford.edu

[4] Secured Socket Layer (SSL)/Transport Layer Security (TLS)



(a) Si-Fi attack (b) Du-Fi attack

Figure 8. Comparison of standard deviation in HTTPS connection setup time between an evil twin AP and a legitimate AP where the evil twin AP utilizes a legitimate AP as it's next hop; (a) shows Si-Fi attack scenario and (b) shows Du-Fi attack scenario.

of τ and k is a tunable factor. After removing noisy data, we supply AP list with corresponding τ values as input to MSC which returns one or more clusters with similar APs grouped together in the same cluster. When the clustering algorithm returns more than one clusters, we identify the scenario as an evil twin AP attack. Thus, MSC returns one cluster in no attack scenarios and two, or more clusters in attack scenarios.

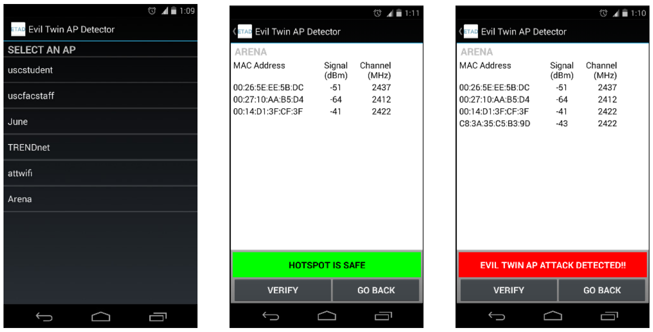
Standard Deviation Analysis. In Section IV-D, we observed that τ values vary significantly in attack scenarios. This indicates that the standard deviation of τ would be larger for an evil twin AP compared to a legitimate AP. On the other hand, the standard deviation would be similar for two legitimate APs in the same hotspot. We depict the standard deviation of τ for every 10 sec interval in Figure 8, where (a) shows the standard deviation for Si-Fi attack, and (b) for Du-Fi attack. As we can see, the standard deviation of τ fluctuates a lot in the attack scenarios compared to regular (no attack) scenarios. Based on this observation, we devised a detection technique where we compare the standard deviation of τ for different APs. In this scheme, we use a threshold γ to detect attack. We calculate the difference of standard deviation between a pair of APs in a hotspot; if difference of standard deviation exceeds the threshold γ, then we identify the case as an attack scenario.

Combined Detection Technique. Both the clustering technique and standard deviation technique can perform independent of each other to detect evil twin AP attacks. However, we use these two techniques in combination in CETAD in order to achieve a higher attack detection rate compared to when these are used independently. We discuss the performance of this technique in Section VI.

C. Security Analysis

The security of CETAD depends on secure data collection and the combination of the three techniques it uses in two phases. The use of HTTPS in the data collection phase ensures that an adversary cannot change source IP address or packet contents, generate a fake message, or shorten τ values to avoid detection. The adversary can at most buffer a packet which will increase τ, but cannot reduce it any way. Thus, adversary cannot evade CETAD by falsifying data. In the following, we analyze security of the detection phase for three scenarios.

1) No Attack: The ISP-based scheme correctly detects no attack scenarios because the ISP information of multiple APs are similar. The timing-based techniques also detect this case correctly due to the similarity in τ values of the legitimate APs.



MAC Address	Signal (dBm)	Channel (MHz)
00:26:5E:EE:5B:DC	-51	2437
00:27:10:AA:B5:D4	-64	2412
00:14:D1:3F:CF:3F	-41	2422

HOTSPOT IS SAFE

MAC Address	Signal (dBm)	Channel (MHz)
00:26:5E:EE:5B:DC	-51	2437
00:27:10:AA:B5:D4	-64	2412
00:14:D1:3F:CF:3F	-41	2422
C8:3A:35:C5:B3:9D	-43	2422

EVIL TWIN AP ATTACK DETECTED!!

(a) Home Screen Regular Scenario Attack Scenario

Figure 9. Screenshots of CETAD app in regular and attack scenarios.

However, there could be false detection in the case when the τ values vary abruptly for two legitimate APs.

2) Mobi Attacks: The ISP-based technique ensures that the adversary will not be able to use his own access network to setup an evil twin AP attack. This also eliminates the possibility that the adversary can use an access network with faster τ compared to the hotspot and thus control the τ values to evade the timing-based technique.

3) Multihop Attacks: In multihop attacks, the adversary sets up Si-Fi attacks or Du-Fi attacks. Since both these attacks use a legitimate AP as the next hop, intuitively she cannot control the value τ to be similar to that of a legitimate AP. On top of the delay from a legitimate AP, the τ value will also depend on the processing delay of software or hardware, Wi-Fi MAC layer collisions, access control, etc. Even if the adversary is able to smooth τ values, CETAD will be able to detect attack using clustering technique. Thus, the adversary cannot manipulate τ values to circumvent CETAD.

VI. Implementation and Results

A. Implementation

We implemented CETAD as an app for Android based phones as a case study. However, it can be implemented for other Wi-Fi devices as well. The app first scans and shows a list of available unique SSIDs. When the user chooses one of the SSIDs from the list, the app creates an AP list with the selected SSID and removes the APs with signal strength below threshold δ, where we used $\delta = -75dBm$. Then for each AP in the list, the app first collects ISP data by using HTTP GET connection to `www.ip2location.com`, and then it collects n instances of τ values by opening an HTTPS connection to a public HTTPS server, e.g., `www.google.com`. We used $n = 10$ in our implementation. Note that some hotspots require users to submit a web form by agreeing to their *terms and conditions*. To automate this step, we use `automatic HTML form submission` technique. However, this technique might fail if the form has protection against automatic submission, e.g., captcha.

After collecting ISP and timing data for all the APs in the list, the app first uses ISP-based detection technique. If ISP-based detection technique cannot detect an attack, then timing-based detection technique is used, which first filters the data using a range of $m \pm k * \sigma$ (Section V-B2). We use $k =$

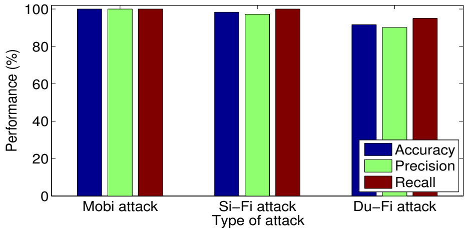


Figure 10. Performance of CETAD for three different types of evil twin AP attacks in several Wi-Fi hotspots in different locations. The hotspots include McDonalds, Starbucks, Wendys, University, etc.

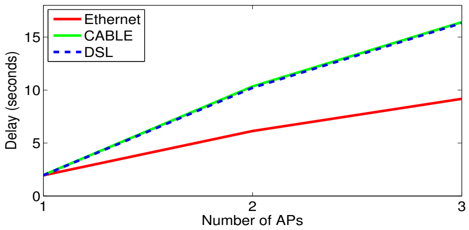


Figure 11. Overhead of CETAD in different types of wireless hotspots. With one AP, our mechanism returns after DHCP discover which indicates that DHCP requires approximately 2 sec for each AP.

2 in our implementation. Then, it uses mean shift clustering algorithm to cluster the τ values for different APs. Apart from the dataset, MSC algorithm requires only one input parameter, window size, h. We used $h = 30ms$ in our implementation. Our algorithm detects an attack when the number of clusters in the input is more than one. Then the mechanism calculates the standard deviation of τ values and applies a threshold value γ to detect an attack. We used $\gamma = 30ms$ in our implementation. When either of the above techniques detects an attack, the app identifies such a scenario as an evil twin AP attack and notifies the user. We depict the screenshots of our app in Figure 9.

B. Experimental Results

Instead of measuring the effectiveness of CETAD in a controlled environment, we launched two types of evil twin AP attack in several hotspots to validate CETAD[5]:

i. **Mobi attacks:** We used a Google Nexus 4 Android phone with 3G data subscription to launch smartphone-based attacks with our own access network, as we discussed in Section III. We used `android-wifi-tether` to setup the evil twin AP for our attack.
ii. **Multihop attacks:** We used a laptop with Windows-7 to launch a Si-Fi attack. We used `Virtual Router version 1.0` for this attack. To launch a Du-Fi attack, we used an additional USB Wi-Fi interface and a laptop with Windows-7. We used `Medialink Wireless-N USB adapter` for our attack.

In each hotspot, we ran CETAD 10 times for each type of the attacks. The hotspot providers include McDonalds, Starbucks, Wendys, University, etc. For the performance analysis, we use the following standard metrics: (a) *Accuracy* indicates how accurately CETAD detects evil twin AP attacks, (b) *Precision* is the fraction of positively detected attacks to all positively detected attacks (correctly or incorrectly), and (c) *Recall* is the fraction of positively detected attacks to all attacks that should be positively detected.

1) Performance: Our experimental results show that CETAD is highly effective in detecting evil twin AP attacks in wireless hotspots as discussed below.

Mobi attacks. CETAD detected Mobi attacks in all cases, i.e., with 100% accuracy, precision, and recall. For this type of attack, our ISP-based scheme was enough to identify the attacks as the ISP information of a legitimate AP, and an evil twin AP were different. For instance, in all cases they had different public IP addresses, zip codes, and usage types.

Multihop attacks. For Si-Fi and Du-Fi attacks, our timing-based scheme was useful because ISP-based scheme cannot detect such attacks as a legitimate AP is used as a gateway by the evil twin AP in these attacks. We depict the performance of CETAD in detecting Si-Fi attacks and Du-Fi attacks in Figure 10 which shows that the accuracy, precision, and recall of our timing-based scheme for Si-Fi attack is 98.33%, 97.22%, and 100.0% respectively; and for Du-Fi attack, it achieved 91.67%, 90.15%, and 95.0% accuracy, precision, and recall respectively.

In summary, CETAD can effectively detect different types of evil twin AP attacks in wireless hotspots.

2) Delay Overhead: Admittedly, our mechanism comes with some delay overhead, since the client application needs to associate with the candidate APs and collect ISP data, and n values of τ. Then after the attack detection phase, we allow the user to associate with a legitimate AP. All these steps incur delay overhead. To analyze the delay overhead, we measure the time difference between the time when the user starts verification and the time when the application shows notification to the user. We depict the delay overhead in Figure 11. When there is only one AP in the AP list, our mechanism does not use the detection scheme because it requires at least 2 APs for detection. In this case, the mechanism returns after DHCP operation is completed. As we can see, the DHCP configuration takes approximately 2 seconds for each AP. As the number of APs increase, the delay overhead increases monotonically. This delay mainly consists of the DHCP delay and delay for collecting n values τ, where $n = 10$. For the three AP scenario, the detection mechanism was complete within 9.2 seconds for Ethernet network and it took approximately 16.4 seconds for DSL network. Despite the small delay overhead, our mechanism is able to detect evil twin AP attacks effectively.

VII. Related Works

In this section, we discuss rogue AP detection schemes in general rather than limiting to evil twin AP detection only. Most of the current rogue AP detection schemes are designed for infrastructure networks, e.g., corporate networks, school, etc. Some of the schemes use a `hybrid approach` requiring custom infrastructure to be installed: Branch et al. [16] designed a Distributed Wireless Security Auditor (DWSA) to

[5] Disclaimer: Several unknown Wi-Fi clients were associated with our evil twin AP during the attack, but we did not record any kind of data from these clients during the attacks.

provide continuous wireless network assessments based on the data from trusted wireless clients; Athawale et al. [14] used mobile agents to quickly scan all possible rogue APs; Bahl et al. [15] designed Dense Array of Inexpensive Radios (DAIR) using commodity USB-based wireless adapters and used an inference engine to detect rogue AP; Ma et al. [30] collected wireless data in promiscuous mode and used AP probing, and OS fingerprint to detect rogue AP. Some other researchers use a `wired approach` for rogue AP detection. This type of approaches monitor wired traffic at the gateway and determine whether the client uses a wired or wireless connection: Wei et al. [31] detects rogue AP by analyzing packet header at the routers and by exploiting the fundamental properties of the 802.11 CSMA/CA MAC protocol as well as the half duplex nature of wireless channels; Beyah et al. [11] and Watkins et al. [13] proposed the use of temporal traffic characteristics to detect rogue APs at a central location (a switch); Shetty et al. [12] identifies unauthorized Wi-Fi hosts connected to rogue APs by analyzing traffic characteristics at the edge (router or gateway) of a network. Both the hybrid and wired approaches require administrative access to the network switch or routers. Additionally, the hybrid approaches require to install additional devices. Hotspot providers are unlikely to utilize these solutions as there is no incentive for them. CETAD does not fall into the infrastructure category because we aim at detecting attack from a client device. However, we can extend CETAD as infrastructure solution as well by deploying client devices with CETAD installed; CETAD can notify the administrators upon detecting an attack.

Only a few works focus on detecting rogue AP at the client end. Gonzales at el. [18] proposed a scheme that first trains a model in a hotspot by creating an AP map based on the user location. Subsequently, the trained model is used to detect evil twin AP by detecting change in the AP map. This scheme cannot detect an evil twin attack if it is ongoing during the training. Moreover, it is dependent on user's location and cannot be used in a new hotspot without training. In comparison, CETAD does not require any training and can be used in any hotspot immediately. Song et al. [17] proposed two algorithms to detect an evil twin AP from the client end. The solutions are based on server Inter-packet Arrival Time (IAT) which is the time interval between two consecutive data packets sent from the server to the client. The solutions require to setup a server within the LAN with their software installed for measuring server IAT and for detecting an evil twin AP. Our work solves a similar problem, but we do not have the rigid requirement of having a custom server inside the LAN, rather we leverage public web server in our mechanism. Additionally, the proposed work cannot detect an evil twin when an adversary uses Mobi attacks, whereas CETAD can detect evil twin AP attacks in various threat models with high detection rate.

VIII. Conclusion

Wi-Fi clients using wireless hotspots are vulnerable to evil twin AP attacks and are in the danger of loosing sensitive information to adversaries. To the best of our knowledge, no mechanism is currently available that can detect evil twin AP attacks without prior training or without installing additional hardware in each hotspot. In this paper, we proposed CETAD, a mechanism that can detect evil twin AP attacks in wireless hotspots and it can be installed in any Wi-Fi enabled device. Our proposed mechanism does not require to install any hardware or software in the hotspot infrastructure. We implemented CETAD as an app for Android-based phones as a case study and showed that it can be used effectively in wireless hotspots to detect evil twin AP attacks.

References

[1] IDC, "Worldwide wlan market reaches nearly 6.4 billion in 2011," in *www.idc.com*, 2012.

[2] R. Myslewski, "Wireless devices to break one-billion barrier in 2011," in *www.theregister.co.uk*, 2011.

[3] JiWire, http://v4.jiwire.com/search-hotspot-locations.htm.

[4] B. Potter, "Wireless hotspots: Petri dish of wireless security," in *Communications of the ACM - Hacking and innovation*. ACM, 2006.

[5] H. Hwang, G. Jung, K. Sohn, and S. Park, "A study on mitm (man in the middle) vulnerability in wireless network using 802.1x and eap," in *Information Science and Security*, 2008.

[6] A. Mishra and W. Arbaugh, "An initial security analysis of the ieee 802.1x standard," University of Maryland, Tech. Rep., 2002.

[7] T. Ohigashi and M. Morii, "A practical message falsification attack on wpa," in *Cryptography and Information Security Conference*, 2009.

[8] Android Wifi Tether, code.google.com/p/android-wifi-tether/.

[9] Virtual Router, virtualrouter.codeplex.com/.

[10] W. Wei, S. Jaiswal, J. Kurose, and D. Towsley., "Identifying 802.11 traffic from passive measurements using iterative bayesian inference," in *INFOCOM*, 2006.

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

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

[13] L. Watkins, R. Beyah, and C. Corbett, "A passive approach to rogue access point detection," in *GLOBECOM*, 2007.

[14] S. Athawale and S. Vanjale, "Detection of rouge access point in 802.11g using ma," *International Journal of Computer Science and Communication*, 2011.

[15] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Enhancing the security of corporate wi-fi networks using dair," in *MobiSys*, 2006.

[16] J. Branch, N. Petroni, L. Van Doorn, and D. Safford, "Autonomic 802.11 wireless lan security auditing," *IEEE Security and Privacy*, 2004.

[17] Y. Song, C. Yang, and G. Gu, "Who is peeping at your passwords at starbucks? to catch an evil twin access point," in *DSN*, 2010.

[18] H. Gonzales, K. Bauer, J. Lindqvist, and D. McCoy, "Practical defenses for evil twin attacks in 802.11," in *GLOBECOM*, 2010.

[19] IEEE, *IEEE 802.11 Wireless Local Area Network*, www.ieee802.org/11/.

[20] IETF, "Dynamic Host Configuration Protocol," in *www.ietf.org/rfc/rfc2131.txt*.

[21] Kevin C. Tofel, *Who has the largest Wi-Fi network in the US? Cable companies say they do*, www.gigaom.com/2013/06/10/.

[22] AT&T, *2.7 Billion Connections Made in 2012; Wi-Fi Network Growth Doubles*, www.att.com.

[23] IETF, "Domain names - implementation and specification," in *www.ietf.org/rfc/rfc1035.txt*.

[24] WiFi Foundation, *Legal*, http://www.wififoundation.org/legal.

[25] Mike Perry, *365-Day: HTTPS Cookie Stealing*, http://fscked.org/talks/ActiveHTTPSCookieStealing.pdf.

[26] Z. Kaleem, "Multiple ssids," in *www.wlanbook.com*, 2008.

[27] A. S. Tanenbaum, *Computer Networks*. Barnes & Noble, 2003.

[28] Internet Assigned Numbers Authority (IANA), www.iana.org.

[29] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.

[30] L. Ma, A. Y. Teymorian, X. Cheng, and M. Song, "Rap: Protecting commodity wi-fi networks from rogue access points," in *Qshine*, 2007.

[31] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. Towsley, "Passive online rogue access point detection using sequential hypothesis testing with tcp ack-pairs," in *Internet Measurement Conference)*, 2007.