2018

# Development of a Client-Side Evil Twin Attack Detection System for Public Wi-Fi Hotspots based on Design Science Approach

Liliana R. Horne
*Nova Southeastern University*, liliana.horne@outlook.com

NSUWorks Citation

Liliana R. Horne. 2018. *Development of a Client-Side Evil Twin Attack Detection System for Public Wi-Fi Hotspots based on Design Science Approach*. Doctoral dissertation. Nova Southeastern University. Retrieved from NSUWorks, College of Engineering and Computing. (1064)
https://nsuworks.nova.edu/gscis_etd/1064.

# Development of a Client-Side Evil Twin Attack Detection System for Public Wi-Fi Hotspots based on Design Science Approach

by

Liliana R. Horne

A dissertation submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy
in
Information Systems

College of Engineering and Computing
Nova Southeastern University

2018

We hereby certify that this dissertation, submitted by Liliana Horne, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

11/14/18

James L. Parrish, Ph.D. — Date
Chairperson of Dissertation Committee

Bennet Hammer — 11 / 14 / 18

Bennet Hammer, Ph.D. — Date
Dissertation Committee Member

11/14/2018

James D. Cannady, Ph.D. — Date
Dissertation Committee Member

Approved:

Meline Kevorkian — 11/14/18

Meline Kevorkian, Ed.D. — Date
Interim Dean, College of Engineering and Computing

College of Engineering and Computing
Nova Southeastern University

2018

An Abstract of a Dissertation Submitted to Nova Southeastern University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

# Development of a Client-Side Evil Twin Attack Detection System for Public Wi-Fi Hotspots based on Design Science Approach

by
Liliana R. Horne
November 2018

Users and providers benefit considerably from public Wi-Fi hotspots. Users receive wireless Internet access and providers draw new prospective customers. While users are able to enjoy the ease of Wi-Fi Internet hotspot networks in public more conveniently, they are more susceptible to a particular type of fraud and identify theft, referred to as evil twin attack (ETA). Through setting up an ETA, an attacker can intercept sensitive data such as passwords or credit card information by snooping into the communication links. Since the objective of free open (unencrypted) public Wi-Fi hotspots is to provide ease of accessibility and to entice customers, no security mechanisms are in place. The public's lack of awareness of the security threat posed by free open public Wi-Fi hotspots makes this problem even more heinous. Client-side systems to help wireless users detect and protect themselves from evil twin attacks in public Wi-Fi hotspots are in great need. In this dissertation report, the author explored the problem of the need for client-side detection systems that will allow wireless users to help protect their data from evil twin attacks while using free open public Wi-Fi. The client-side evil twin attack detection system constructed as part of this dissertation linked the gap between the need for wireless security in free open public Wi-Fi hotspots and limitations in existing client-side evil twin attack detection solutions. Based on design science research (DSR) literature, Hevner's seven guidelines of DSR, Peffer's design science research methodology (DSRM), Gregor's IS design theory, and Hossen & Wenyuan's (2014) study evaluation methodology, the author developed design principles, procedures and specifications to guide the construction, implementation, and evaluation of a prototype client-side evil twin attack detection artifact. The client-side evil twin attack detection system was evaluated in a hotel public Wi-Fi environment. The goal of this research was to develop a more effective, efficient, and practical client-side detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi hotspots. The experimental results showed that client-side evil twin attack detection system can effectively detect and protect users from mobile evil twin AP attacks in public Wi-Fi hotspots in various real-world scenarios despite time delay caused by many factors.

## Acknowledgements

First and foremost, I would like to thank God for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily.  Without God's blessings, this achievement would not have been possible.

I thank my Dissertation Chairperson, Dr. James Parrish, and committee members, Dr. James Cannady and Dr. Bennet Hammer, for their patience, guidance, and support throughout this dissertation process.

Additional thanks go to my family.  The love and blessings of my parents Lilian Argüello de Ribadeneira and Wolfio Ribadeneira and encouragement of my admirable brother and sister, Alex and Gigi, have all made a tremendous contribution in helping me reach this stage in my life.  This would not have been possible without your unwavering and unselfish support given to me at all times.

I also would like to express my gratitude to my extended family.  The love and support of my mother-in-law Ann Horne and father-in-law Kermit Horne, my sisters and brothers Patti, Cathy, Susan, Ray and Billy.  I thank them for supporting me on to follow my dream of getting this degree.

Finally, I would like to dedicate this work to my loving husband Joe Horne and brilliant daughter Angie Horne.  The biggest source of my strength.  Without your amazing love and support, I would not be where I am today and what I am today.  You both have been such an inspiration throughout this journey.  Thank you for your sacrifices and support specially during the past four years as I worked towards the completion of this research.

This is for you future President, Inventor, and Doctor Angie!

## Table of Contents

**Appendices**

## List of Tables

**Tables**

# List of Figures

## Figures

# Chapter 1

# Introduction

## Background

The use of IEEE 802.11 based Wi-Fi or wireless local area networks (WLANs) has grown to become the predominant method of access to the Internet in the last few years (Hossen & Wenyuan, 2014). Mobile users can have Internet access anywhere there is service and anytime there is not an outage. Public places, such as hotels, restaurants, cafes, airports and others have made open (unencrypted) Wi-Fi Internet access available at no cost to either attract customers or better serve current customers. Locations that provide open and free Wi-Fi Internet access are called public Wi-Fi hotspots. According to JiWire Mobile Audience Insights Report (2013), Internet access via public Wi-Fi networks has become widely available and largely free of charge, with over 81 percent of all public Wi-Fi hotspots offering free connections as an alternative to paid. Additionally, user demand for free, high speed Internet connections is growing rapidly as mobile devices that require higher bandwidth continue to increase. Simultaneously, the number of public Wi-Fi hotspots are expanding. Industry research overwhelmingly demonstrates that Wi-Fi is now the preferred free-access technology for travelers' mobile devices. Worldwide public Wi-Fi hotspot deployments have reached a total of 5.69 million in 2014 and will grow at a Compound Annual Growth Rate (CAGR) of 11.2% between 2015 and 2020. This includes public Wi-Fi hotspots deployed by mobile and fixed-line carriers as well as third-party Wi-Fi service providers. ABI Research expects the number of worldwide carrier Wi-Fi hotspots will reach 13.3 million in 2020 (ABI Research, 2015).

Han, Sheng, Tan, Li, and Lu (2009, 2011) indicated that with the increase of users who come to expect free wireless availability, the security of such networks becomes increasingly more important. According to a survey by Private Wi-Fi in partnership with The Identity Theft Resource Center (ITRC) (2013), U.S. consumers are three times more likely to connect to a Wi-Fi network if it is free. The ITRC calls this trend “The Convenience Factor”, which describes the fact that Wi-Fi hotspots are available in many public places, which affords users the ability to get and stay connected at no cost, wherever they are.

Additionally, Kim, Park, Jung, and Lee (2012) and Nikbakhsh, Zamani, Abdul Manaf, and Janbeglou (2012) stated in their studies that the growing popularity of WLANs, increases the risk of wireless security attacks. Since the goal of free open public Wi-Fi hotspots is to provide convenience and to attract customers, no security tools are in place. For instance, most public Wi-Fi hotspots provide free, open, and zero liability Internet access to customers (Hossen and Wenyuan, 2014). Wi-Fi’s popularity makes it an attractive target for attackers to access and capture wireless client information. For a wireless user, it is impossible to determine the safety of an open public Wi-Fi hotspot and identify the ones that are dangerous. Unfortunately, Wi-Fi users have to take responsibility for their own security when connecting to free open public Wi-Fi networks.

While users are able to access free open Wi-Fi Internet hotspot connections in public more conveniently, they are more vulnerable to a particular type of fraud and identity theft, referred to as evil twin attacks (ETA). An evil twin attack in a wireless LAN is a reference to a hard- or software-based 802.11 rogue Wi-Fi access point (AP) that looks like a legitimate one offered on the premises, but actually has been set up by a hacker to eavesdrop all wireless communications done by the victims.  Evil twin attacks can significantly threaten the security of wireless users of public Wi-Fi hotspots (Song, Yang, and Gu, 2010, 2012; Hossen and Wenyuan, 2014; Nakhila, Dondyk, Amjad, and Zou, 2015; Hsu, Wang, Hsu, Cheng, and Hsneh, 2015).  Moreover, lack of knowledge and awareness possessed by the Wi-Fi hotspot users make this issue extremely disturbing (Nikbakhsh et al., 2012).  Many Wi-Fi hotspot users are oblivious to the hidden risks that the technology poses, such as identity theft, hacking, and stolen bank accounts. Due to its gravity, the evil twin attack has gained a notable interest in the media and research community (Han et al., 2009, 2011; Song et al., 2010, 2012; Lanze, Ponce-Alcaide, Panchenko, and Engel, 2014).

The detection of ETA has been researched for many years. Researchers have been investigating detection methods that can alert the wireless network administrator or the user about the presence of this type of attack.  Song et al. (2010, 2012) found that existing evil twin detection solutions are mainly for network administrators (administrator-side) instead of for a wireless client or user (client-side) to detect an evil twin attack. According to Song et al. (2010, 2012) and Hsu et al. (2015), administrator-side solutions are expensive, limited by requiring the knowledge of authorization users and AP list, hardly maintained, difficult to protect users timely when the attack is launched, and not available for many cases.

Kim et al. (2012) indicated that administrator-side methods utilize extra devices, sometimes referred to as Wireless Intrusion Detection System (WIDS) nodes. The WIDS nodes monitor the wireless traffic and route the gathered traffic to their servers. The servers get to know the wireless environment in order to detect evil twin APs using wireless traffic from WIDS nodes. However, if a user moves to other locations where there are no WIDS nodes, the administrator-side methods can no longer assure secure communication in WLANs for mobile users. Although there are WIDS nodes for secure communication, the detection methods hardly detect the evil twin access points (APs) when the servers have not yet learned the wireless environments. In support of Kim et al.'s (2012) study, Hossen and Wenyuan (2014) found that businesses offer public Wi-Fi hotspots to provide free Internet service to attract customers. They have little motivation to pledge secure Internet surfing or to setup more devices or install detection hardware and software in their infrastructure to detect an evil twin access point (AP) attack. In addition, Hossen and Wenyuan (2014) found that administrator-side solutions are not applicable to public Wi-Fi hotspots and more practicable in environments such as infrastructure networks, e.g. corporate networks. In public Wi-Fi networks, wireless users should not assume that the network provider will deploy any type of security protections against evil twin attacks. Furthermore, according to Nakhila et al. (2015), administrator-side detection solutions will add more cost to the total wireless network construction price. This is because network administrators need to implement wireless devices that act as wireless sensors to continuously scan the airwaves and gather information about the transmitting APs.

Monica and Ribeiro (2011) found that administrator-side solutions are not real-time, allowing short-term evil twin attacks to remain unnoticed. Additionally, even if the detection is done in a timely manner, many users can still be victims of the attack, since there is no automatic way of denying access to the evil twin APs or even to advise users of the attack.  Additionally, Song et al. (2010, 2012) and Monica and Ribeiro (2011) indicated that administrator-side approaches still have the risk of falsely claiming a normal neighbor AP as a rogue AP with a high probability.

To address this problem, Song et al. (2010, 2012) suggested that traveling users who use wireless networks at free open public hotspots need to protect themselves from evil twin attacks, instead of having any reliance on the providers of free open public Wi-Fi hotspots, which typically do not provide security for public Wi-Fi hotspot users. Song et al. (2010, 2012) claimed that a lightweight and effective client-side solution for traveling users is highly desirable.  According to Nikbakhsh et al. (2012) and Kim et al. (2012), client-side solutions help the user of public Wi-Fi (who neither has an AP authorization list nor any sophisticated software or hardware) to independently determine whether an AP is legitimate or not without any help from network administrators.  Public Wi-Fi users are vulnerable to big security risks such as connecting to a hacker's rogue access point.  This is due to users not having prior knowledge of the public Wi-Fi hotspot's network they are connecting to.  Rogue access points expose wireless users to evil twin attacks in which the hacker can capture all the user's network traffic.  Nakhila et al. (2015) further indicated that client-side detection is more appropriate than administrator-side detection since it gives security-sensitive users more control over their Wi-Fi connection security.

Monica and Ribeiro (2011) found that public Wi-Fi hotspots are beneficial for wireless users as well as service providers that wish to attract clients.  However, under evil twin attacks, the wireless user innocently associates to an attacker's wireless access point and the attacker proceeds to compromise user's sensitive information.  According to Monica and Ribeiro (2011), client-side detection solutions that are efficient and effective are in great need.

Kim et al.’s (2012) study indicated that recently several evil twin AP detection methods have been designed in order to overcome the administrator-side problems on the client-side. However, most of the existing client-side solutions only target multihop attacks where the attacker uses a legitimate AP for accessing the Internet to pass through client’s data (Nakhila et al., 2015). According to Nakhila et al. (2015), these detection methods will fail when the attacker launches a mobile attack which uses a different gateway compare with the legitimate AP. Evil twin attacks that use their mobile Internet (mobile attacks) will become more popular nowadays due to the increase in the Internet access speed of mobile connections, such as 3G/4G Long Term Evolution (LTE). Additionally, in support of Nakhila et al. (2015), Szongott, Henne, and Smith (2012) and Hossen & Wenyuan (2014) indicated that the inclusion of mobile hotspot capabilities in virtually all new mobile devices opens the door to mobile evil twin attacks. Unfortunately, there is limited research focused on client-side solutions that will allow wireless users to verify the authenticity of access points at free open public Wi-Fi hotspots and protect themselves from mobile evil twin attacks. Additionally, the client-side detection solutions proposed so far have limitations regarding requirements, assumptions, and evaluation approaches.

## Problem Statement

The problem explored in this dissertation report is the need for client-side detection solutions for wireless users to be able to protect themselves from evil twin attacks while using free open public Wi-Fi.  Existing literature mainly focus on client-side evil twin attack detection methods for multihop attacks.  These detection methods will fail when the attacker launches a mobile attack.  Mobile evil twin attacks will become more popular nowadays due to the increase in the Internet access speed of mobile connections and the inclusion of mobile hotspot capabilities in virtually all new mobile devices (Szongott et al., 2012; Hossen & Wenyuan, 2014; Nakhila et al., 2015).  Unfortunately, there is limited research focused on client-side evil twin attack detection solutions for mobile attacks.  Additionally, existing solutions have limitations regarding requirements, assumptions, and evaluation approaches.  As a result, wireless users of free open public Wi-Fi hotspots are vulnerable to mobile evil twin attacks in which the attacker can intercept, collect, and manipulate user’s sensitive data.

The problem exists due to the lack of more effective, efficient and practical evil twin attack detection systems for mobile attacks on the client side.  According to Hossen & Wenyuan (2014) and Szongott et al. (2015), existing client-side detection solutions are impractical and thus have not seen any adoption.  As the literature in proceeding paragraphs and chapters will reveal, there are two types of evil twin attack scenarios (Song et al., 2010, 2012; Nikbakhsh et al., 2012; Hossen & Wenyuan, 2014; Nakhila et al., 2015).  The first scenario is when the attacker uses the legitimate AP for Internet access.  In this scenario, the evil twin AP can itself behave as a normal Wi-Fi client and uses the legitimate AP to connect to the Internet.  All the wireless traffic from the victim will pass through the attacker’s node.  In the literature, this type of attack is denoted as multihop attack.  The second scenario is when the attacker uses mobile Internet (e.g. 3G/4G LTE) as the access network for connecting to the Internet.  In this scenario, the evil twin AP uses a different gateway compared with the legitimate AP. A hotspot router can act as an evil twin AP. Also, a smartphone with mobile AP functionality built in operating systems such as Android or iOS, can act as an evil twin AP and the setup is trivially easy. In the literature, this type of attack is denoted as mobile attack.

Most of the existing client-side evil twin detection methods fall under the first scenario. Han et al. (2009, 2011), Song et al. (2010, 2012), Monica & Ribeiro (2011), Nikbakhsh et al. (2012), Kim et al. (2012), Lanze et al. (2014), and Hsu et al. (2015) assume in their studies that the attacker uses the legitimate wireless network gateway to pass through client data traffic (multihop attacks). However, their detection methods will fail when the attacker uses a different gateway (mobile attack) with a faster Internet connection compared to the legitimate wireless network (Nakhila et al., 2015). Additionally, according to Szongott et al. (2012), Hossen & Wenyuan (2014), and Nakhila et al. (2015), mobile attacks will become more popular nowadays due to the increase in the Internet access speed of mobile connections and the inclusion of mobile hotspot capabilities in virtually all new mobile devices.

Han et al. (2009, 2011) developed a client-side timing-base method for detection of rogue access points based on round-trip time (RTT) calculation between the wireless user and the DNS server, and does not require administrator assistance. Their RTT-based method helps distinguish the route through a rogue AP from that through a legitimate AP (one hop versus two-hop wireless channels). However, the issue with timing-based methods is that with the increase in wireless networks speeds, transmission delay differences between a wireless node and a wired node will eventually fade. This means that a multihop setting may become indistinguishable from a one-hop setting (Monica &

Ribeiro, 2011). Timing-based detection will be unreliable when the attacker uses a faster Internet connection as the evil twin AP (Nakhila et al., 2015). Also, their approach utilizes the training detection technique which requires pre-gathering the information of the target wireless network. This method could not be applied to public Wi-Fi users at the client side, since once users are in different areas, the network situation may have significantly changed. The trained knowledge in one wireless network can be hardly applicable to another network (Song et al., 2010, 2012).

Song et al. (2010, 2012) also developed a client-side timing-base method called "ETSniffer" (Evil Twin Sniffer) based on Interpacket Arrival Time (IAT) to detect evil twin access points by distinguishing a one-hop from a two-hop wireless network setting between the wireless client and the remote IAT server (custom server). Their method does not require administrator assistance. However, their method requires setting up additional equipment such as a custom server within the LAN with their software installed for measuring server IAT and for detecting an evil twin AP. According to Han et al.' (2009, 2011), Kim et al.' (2012), Nikbakhsh et al.' (2012), Lanze et al.' (2014), Hossen & Wenyuan' (2014), and Nakhila et al.' (2015) studies, to guarantee usability and availability to the client, a client-side detection method must discover evil twin APs using their Wi-Fi enabled devices (e.g. laptops, smartphones, tablets, etc.) without any additional equipment.

Monica & Ribeiro (2011) developed a client-side evil twin detection system called "WiFiHop". Their method does not require network administrator assistance. This detection system is based on the behavior of the legitimate AP without depending on timing to detect a multihop setting between the wireless user and the Internet. However, their solution requires the implementation of an echo-server deployed through the use of a script on any public hosting server; therefore, requiring hotspot network modification. According to Hossen & Wenyuan (2014), a client-side detection solution must be able to verify an access point in a hotspot and thus cannot assume any custom infrastructure support. Further, Hossen & Wenyuan stated that designing an infrastructure-side solution would require hotspot providers to re-design existing hotspots, which is unlikely to happen because most hotspots are free services with no independent revenue.

Nikbakhsh et al. (2012) developed a client-side approach based on traceroute that compares the gateways and routes that a packet travels to determine whether an access point is legitimate or not. Their method does not require administrator assistance. However, according to Nakhila et al. (2015), the attacker can capture traceroute results transmitted to the wireless client using the legitimate wireless network and convey those results to the wireless client by means of the rogue wireless network. This will give the same route information for both gateways. Also, as mentioned previously this method provides limited client-side detection targeted only to the specific evil twin attack scenario where the attacker uses the legitimate AP for Internet access instead of a more popular scenario where the attacker uses mobile Internet as the access network for connecting to the Internet (Nakhila et al., 2015). Lastly, Nikbakhsh et al.’s (2012) approach was not implemented or evaluated in a lab environment or in the field. Therefore, no conclusions can be drawn of its effectiveness.

During the same time, Kim et al. (2012) developed a client-side evil twin attack detection method for smartphones based on received signal strengths (RSSs), and does not require administrator assistance. Their method measured RSSs from both the legitimate and evil twin access points on the smartphone and used normalization of collected signal strengths for accurate measurement.  Highly correlated RSSs are considered fake signals from an evil twin access point.  However, their method was also based on the attacker using the legitimate wireless network gateway to pass through client data traffic.  In addition, Kim et al.' (2012) method only works with smartphones associated with a mobile communication network.

Lanze et al. (2014) developed a client-side method for detection of evil twin attacks operated by software.  Their method does not require administrator assistance.  Their method separates software access points from legitimate hardware access points.  However, their approach was only evaluated in a lab environment.  Therefore, no final conclusions can be drawn on its effectiveness.

Hsu et al. (2015) proposed a client-side evil twin attack detection system called "ET Detector" based on redirection behavior, and does not require administrator assistance.  By operating the wireless network interface controller (WNIC) in monitor mode (which is able to capture all packets that conform to its monitoring channel and protocol) and through analyzing the captured packets, users can easily and precisely detect evil twin attacks.  However, the system has two detection mechanisms: default testing and secondary device testing.  Default testing only works when a user is not the only one using public Wi-Fi in a hotspot.  Otherwise, the system will be forced to use secondary testing which requires an extra Wi-Fi device with no sensitive data on it to associate to the target AP to make the detection.  Therefore, the system is not automated and requires intervention from users.  According to Han et al.' (2009, 2011), Monica & Ribeiro' (2011), Kim et al.' (2012), Nikbakhsh et al.' (2012), Lanze et al.' (2014), Hossen &

Wenyuan' (2014), and Nakhila et al.' (2015) studies, to guarantee usability and availability to the client, a client-side detection method must discover evil twin APs without additional equipment. Also, Monica & Ribeiro (2011), as well as Kim et al. (2012), Hossen & Wenyuan (2014), Nakhila et al. (2015), and Szongott et al. (2015) indicated that the client-side evil twin detection system must be automated with no intervention from users. Lastly, this method provides limited client-side detection targeted only to the specific evil twin attack scenario where the attacker uses the legitimate AP for Internet access.

Szongott et al. (2015) proposed a detection system called Mobile Evil Twin Detection System (METDS) for smartphones based on context-based recognition, which uses as much environmental data of smartphones as possible during the association process to help decide if the access point is legitimate or the user needs to be warned of a potential attack. Their method does not require administrator assistance. However, their method requires previous knowledge of the network in order to assist the user and also an external server to store learned data. Studies conducted by Hossen & Wenyuan (2014) and Nakhila et al. (2015), indicated that to detect an evil twin AP, the system should not require any training knowledge of the target wireless network. Also, according to Han et al.' (2009, 2011), Kim et al.' (2012), Nikbakhsh et al.' (2012), Lanze et al.' (2014), Hossen & Wenyuan' (2014), and Nakhila et al.' (2015) studies, to guarantee usability and availability to the client, a client-side detection method must discover evil twin APs using their Wi-Fi enabled devices (e.g. laptops, smartphones, tablets, etc.) without any additional equipment. Lastly, their method only works with smartphones associated with a mobile communication network.

As the literature review in proceeding paragraphs and chapters will reveal, Nakhila et al. (2015) and Hossen & Wenyuan (2014) are the only existing studies that assume the attacker using a different gateway from a legitimate AP (mobile attacks). Nakhila et al. (2015) presented a client-side detection method for mobile attacks that detects whether or not different gateways are used by multiple APs in one hotspot location that have the same SSID. Their method does not require administrator assistance. Their detection technique relies on an SSL/TCP connection to a remote public server, and detects the changing of wireless network gateway's public IP address in the middle of the SSL/TCP connection. However, Nakhila et al.'s method does not take into account that the attack can be executed before the client establish a secure connection to the remote server. Additionally, Nakhila et al. assume that the BSSID (MAC address) of the hotspot APs is unique and use that as a reference in their method to switch between different APs with the same SSID in the hotspot. Nakhila et al. did not assume the scenario when the attacker uses the same SSID and BSSID of a hotspot legitimate AP. According to Szongott et al. (2015) and Kumar and Paul (2016), SSIDs and BSSIDs can easily be spoofed by an attacker as the legitimate APs always transmit the SSIDs and the BSSIDs. Furthermore, Nakhila et al. did not cover the scenario when the attacker blocks access to the public website. Nakhila et al. assume that all the hotspot APs with the desired SSID are detected during the initial wireless network scanning and that the client is able to associate to all the APs in the public Wi-Fi network, which in practice is not always the case. Nakhila et al.'s (2015) method only works when the mobile evil twin AP is in the same subnet as the legitimate AP. Lastly, their approach was only evaluated in a lab environment. Therefore, no final conclusions can be drawn on its effectiveness.

Furthermore, Hossen & Wenyuan (2014) introduced a method called Client End Evil Twin Access Point Detector (CETAD) to detect evil twin attacks, and does not require administrator assistance. Their detection technique relies on a public server. Their application included two detection techniques: ISP-based and timing-based. The application utilized the ISP-based detection technique, and if not successful, used the timing-based detection technique. The ISP-based technique was used to detect mobile attacks as the ISP information of a legitimate AP and an evil twin AP are different. Similar to Nakhila et al.'s method, it detects whether or not different gateways are used by multiple APs in one hotspot location that have the same SSID. Timing-based technique was used to detect multihop attacks because the attacker's evil twin AP uses the legitimate AP as the gateway. However, as stated previously, timing measurements are technology dependent (Monica & Ribeiro, 2011). Also, Hossen & Wenyuan's assume that the BSSID of the hotspot APs is unique and use that as a reference to switch between different APs with the same SSID in the hotspot. Hossen and Wenyuan did not assume the scenario when the attacker uses the same SSID and BSSID of a hotspot legitimate AP. Hossen & Wenyuan's assumed that the mobile twin AP is in a different subnet as the legitimate AP. Furthermore, Hossen & Wenyuan assumed that all the hotspot APs with the desired SSID are detected during the initial wireless network scanning and that the client is able to associate to all the APs in the public Wi-Fi network, which in practice is not always the case. Lastly, Hossen & Wenyuan's ISP-based method for mobile attacks uses a public website to gather the global IP address shared by the legitimate APs. However, Hossen & Wenyuan's method does not cover the scenarios when the attacker blocks access to the public website or when the attacker presents an invalid certificate while ISP information is retrieved from the public website.

Hossen & Wenyuan's (2014) and Nakhila et al.'s (2015) evil twin attack detection methods for mobile attacks do not protect the public Wi-Fi users for the duration of the Wi-Fi connection, discovering and reporting on new mobile evil twin access points. Protection is only provided to the user at the beginning based on the assumption that the attacker with a mobile evil twin AP will be in the hotspot when the user initially runs the detection system. In a real life environment, an attacker may not be present when the user connects to the hotspot. An attacker with an evil twin AP could arrive at the public Wi-Fi hotspot at a later time.

Additionally, all existing client-side approaches assumed that the client has not connected to the target public Wi-Fi network in the past. According to Kumar & Paul (2016), the operating system stores the SSID and BSSID with which it was previously connected to in the client's preferred network list, and it is always in the exploration of the same and whenever detects attempts to connect to it. Therefore, the client will automatically connect to a potential evil twin AP when using the public Wi-Fi hotspot. Also, existing client-side detection systems only warn the user of the presence of an evil twin AP. After detection, they do not allow the client to connect to a legitimate AP to access the Internet. Specifically, in regards to mobile attack approaches, Nakhila et al.'s method is not able to identify which AP is rogue and which one is legitimate arguing that since both the legitimate AP and the rogue AP provide Internet access that could have similar quality, it is very difficult to further tell them apart. In addition, Hossen & Wenyuan (2014) claim that after the attack has been detected, the system allows the wireless user to connect to the legitimate AP; however, this was not included in their algorithm. Finally, existing studies used their own mobile evil twin APs on their lab and field evaluations. They did not aim at detecting real mobile evil twin APs in the wild.

The client-side detection system constructed as part of this dissertation linked the gap between the need of wireless security in free open public Wi-Fi hotspots and limitations in existing client-side evil twin attack detection solutions.

## Dissertation Goal

The goal of this dissertation was to develop a more effective, efficient, and practical client-side evil twin attack detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi hotspots. To resolve the problem statement above, the author focused on developing a client-side evil twin attack detection system for users of free open public Wi-Fi hotspots based on the following requirements gathered from the literature review:

1. It protects users from attackers that use a different gateway from a legitimate AP (mobile attack).
2. It protects users whether or not they have connected to a free open public Wi-Fi network in the past.
3. It protects users when not all the hotspot APs with the desired SSID are detected during the initial wireless network scanning.
4. It protects users when the client is not able to associate to all the APs in the public Wi-Fi network.

5. It protects users when the attacker sets up the mobile evil twin AP with the same SSID, BSSID (MAC address), and subnet of a legitimate AP.
6. It protects users when the attacker blocks access to the public website used to get ISP information.
7. It protects users when the attacker presents an invalid certificate while retrieving ISP information from a public website.
8. After mobile evil twin AP attack detection, it connects the user to a legitimate AP.
9. It protects users for the duration of the public Wi-Fi connection, discovering and reporting on new mobile evil twin access points.
10. It is evaluated in the wild aiming to detect real mobile evil twin APs. In case of not detecting real mobile evil twin APs during the field evaluation period, it is evaluated with the mobile evil twin AP used in the lab.
11. It is not based on timing or traceroute.
12. It does not require any additional equipment.
13. It does not require modification of the hotspot network infrastructure (custom infrastructure support).
14. It does not require trained knowledge of the target wireless hotspots infrastructure.
15. It is automated with no intervention from users.

In support of the goal, this study leveraged DSR literature, Hevner, March, Park, and Ram (2004) seven steps of effective DSR, Peffers, Tuunanen, Rothenberger, and Chatterjee’s (2008) DSR Methodology (DSRM), Gregor and Jones (2007) IS Design Theory, and Hossen & Wenyuan’s (2014) study evaluation methodology to promote guiding design principles, procedures and specifications for the construction, implementation and evaluation of the prototype client-side evil twin attack detection artifact.

## Research Questions

The research questions identify the specific objectives this dissertation report addressed and helped shape the conceptual framework for the study. This study focused on the design, development, and evaluation of a client-side evil twin attack detection system for public Wi-Fi users to protect themselves from mobile evil twin attacks and answered the following questions:

| Peffers, Tuunanen, and Rothenberger (2008) - DSRM Activity | Research Questions |
|---|---|
| Define the objectives for a solution | 1. What requirements must the product meet in order to address the problem? |
| Design & Develop | 2. What are the major decision points in the design and development process? |
| Demonstrate & Evaluate | 3. In what way does the product developed meet and fail to meet the requirements specified? |

## Relevance and Significance

The problem in this dissertation is both meaningful and research-worthy since connecting to public Wi-Fi hotspots leaves users vulnerable to evil twin attacks from hackers. According to the Identity Theft Resource Center (2013), an evil twin is the wireless version of a “phishing” scam: an attacker tricks wireless users into connecting their mobile devices by impersonating as a legitimate access point to eavesdrop on wireless communications. Wireless users make the assumption that using a Wi-Fi hotspot at a hotel or at an airport is no different than logging into the network at home or at the office. Business travelers willing to connect to public Wi-Fi networks that provide free Internet access are specifically vulnerable to evil twin attacks. It is impossible to tell the safe networks from the bad ones. Wireless eavesdropping can occur anywhere. Many public Wi-Fi hotspots pass responsibility entirely to wireless users for their mobile device security.

Challenges exist in tracing a hack that occurs on a free open public Wi-Fi network. Song et al. (2010, 2012) and Monica & Ribeiro (2011) indicated that evil twin attacks can be hard to trace. The attacker can shut off the attacks suddenly or randomly after accomplishing the malicious goals. In a very short time frame, the attacker may already have compromised public Wi-Fi user's sensitive information, such as passwords or credit card information. Nevertheless, Norton Cybercrime Report (2011) stated that over three quarters 77% of those who use free open public Wi-Fi have experienced cybercrime in comparison to 62% of those who do not.

A study conducted by The Guardian in 2011, launched two evil twin attacks conducted with volunteers, in which they successfully gather users' usernames, passwords, messages and even credit card information. This study reinforced that many public Wi-Fi hotspots have no forms of identification, except their wireless network names (SSID), which can be easily impersonated. Additionally, one recent study from Private Wi-Fi (2011) found that over 56% of laptops were broadcasting the name of their trusted Wi-Fi networks, and that 34% of them were willing to connect to unsecure public Wi-Fi networks. Consequently, to quantify the scale of the threat of evil twin attacks on victims, Szongott et al. (2015) completed a field study with 92 participants to gather their Wi-Fi usage patterns. With this data, Szongott et al. (2015) revealed the number of participants potentially exposed to an evil twin attack. The authors collected data from 223,877 connections that were initiated to access points during the study and gathered anonymous statistics about all configured networks on the participants' devices. Figure 1 shows the amount of configured wireless networks per user. They are differentiated by unencrypted networks like open (unencrypted) public access points and encrypted networks, that use encryption schemes like WPA2 (Szongott et al., 2015). In total Szongott et al. (2015) gathered data about 239 open (unencrypted) Wi-Fi networks from all of the 92 participants' devices. The study demonstrates that a significant number of users are exposed to evil twin attacks and that the mobile devices automatically initiated most connections to popular open access points without the user being aware of the connection (Szongott et al., 2015).

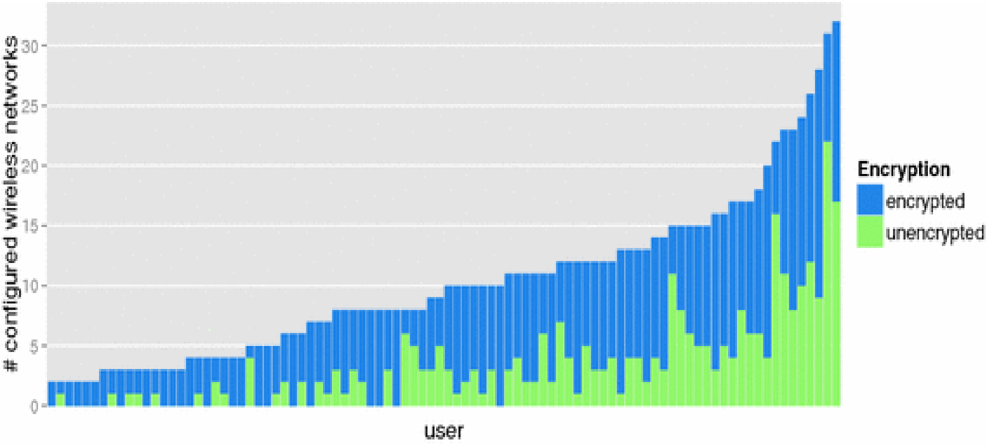


*Figure 1.* Number of configured Wi-Fi networks on participants' devices, divided into unencrypted (green) and encrypted (blue) networks.

To strengthen the relevance of this problem, there are a number of academic studies supporting the argument that client-side evil twin attack detection architecture supports public Wi-Fi hotspots, improves public Wi-Fi user security, and is significant. Song et al. (2010, 2012) and Hsu et al. (2015) indicated that existing evil twin attack detection solutions are mostly for wireless network administrators instead of for a wireless client to detect an evil twin attack at public Wi-Fi hotspots. The researchers also indicate that administrator-side solutions are expensive, limited by requiring the knowledge of authorization users and AP list, hardly maintained, difficult to protect users timely when the attack is launched, and not available for many public hotspots scenarios. Additionally, Song et al. (2010, 2012) indicate that traveling users who use public Wi-Fi hotspots need to protect themselves from evil twin attacks and that a lightweight and effective client-side solution for these users is highly desired.

Furthermore, Nikbakhsh et al. (2012) indicate that public Wi-Fi users are vulnerable to security risks such as connecting to a hacker’s rogue access point. Users do not have prior knowledge of the public Wi-Fi hotspot’s network they are connecting to and usually connect to the wireless access point with the best signal strength. Further, Nikbakhsh et al. (2012) found that rogue access points expose wireless users to evil twin attacks in which the hacker can capture all the user’s network traffic and that wireless users lack of knowledge of this security issue. Nikbakhsh et al. (2012) indicate that client-side methods such as the client-side artifact proposed in this study will need to be designed and constructed to warn wireless users to connect to rogue access points in public Wi-Fi hotspots.

Additionally, Hossen and Wenyuan (2014) found that there is no security authentication mechanism of Wi-Fi access points available in open public Wi-Fi hotspots, which makes wireless users vulnerable to evil twin attacks. This type of attack allows a hacker to steal sensitive data from wireless users. Currently, there is not a method that will allow a user to verify the integrity of an access point at wireless hotspots. Consequently, the relevancy of evil twin attack detection solutions using client-side architecture is evident and supports the primary driver for advancing the research through this dissertation report.

According to Hossen & Wenyuan (2014) and Szongott et al. (2015), existing client-side detection solutions are impractical and thus have not seen any adoption. Most of existing client-side solutions protect users from multihop attacks. These detection methods will fail when the attacker launches a mobile attack (Nakhila et al., 2015). According to Szongott et al. (2012), Hossen & Wenyuan (2014), and Nakhila et al. (2015), mobile evil twin attacks will become more popular nowadays due to the increase in the Internet access speed of mobile connections and the inclusion of mobile hotspot capabilities in virtually all new mobile devices. Unfortunately, there is limited research focused on client-side solutions for mobile attacks. Additionally, existing solutions have limitations regarding requirements, assumptions, and evaluation approaches. As a result, wireless users of public Wi-Fi hotspots are vulnerable to mobile evil twin attacks in which the attacker can intercept, collect, and manipulate user’s sensitive data. To address the research problem, this study developed a more effective, efficient, and practical client-side evil twin attack detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi hotspots.

Finally, Hevner (2007) stated that DSR is essentially pragmatic in nature due to its emphasis on relevance; making a clear contribution into the application environment. The relevance cycle initiates DSR with an application context that not only provides the requirements for the research as inputs but also defines acceptance criteria for the ultimate evaluation of the research results. Therefore, deriving artifact and process building to facilitate the construction and evaluation of the client-side evil twin attack detection architecture based on DSR developed in this study made a clear contribution to problems that span public Wi-Fi.

## Barriers and Issues

Despite the fact that the equipment, network, and facilities are accessible to design, construct, and evaluate the effectiveness of the prototype production system, challenges exist in evaluating the system using public Wi-Fi users (as users of the detection system) and at a large scale. In order to analyze and evaluate the artifact using public Wi-Fi users and at a large scale, the system would need to be made available to a large number of actual traveling users who can test the system in many public Wi-Fi locations for a defined period and report back to the researcher on detection effectiveness and efficiency. Furthermore, before the study, the author would need to instruct the actual public Wi-Fi users on how to operate the system, and at that point, the study would contain bias, because the author would have made the users more attentive to security risks related to an evil twin attack.

The client-side evil twin attack detection artifact reported in this study aimed at detecting mobile evil twin attacks in the wild. Since no mobile evil twin APs were detected in the wild, the author proceeded to evaluate the artifact with the mobile evil twin AP used in the lab. The system was designed and developed based on DSR with the objective of principally addressing the study research questions. The system performance was evaluated extensively at a public Wi-Fi hotspot and using a researcher-participant approach. The author received consent from the public Wi-Fi hotspot to perform the evaluation. This is a requirement even when the evaluation is performed in public Wi-Fi hotspots.

## Assumptions, Limitations, and Delimitations

The first assumption made in this study is that, since the artifact was going to be evaluated in the wild at a hotel that offered free open public Wi-Fi, attackers were going to perform mobile evil twin AP attacks in the hotel public Wi-Fi hotspots during the evaluation time period. The second assumption was that the attacker was going to use his smartphone with mobile AP functionality to launch an evil twin AP attack (mobile attack). Since no mobile evil twin APs were detected in the wild during the field evaluation period, the author proceeded to evaluate the artifact with the mobile evil twin AP used in the lab.

The first limitation is that since the artifact was evaluated in the wild, the researcher could not control when the attackers would appear at the hotel to perform the evil twin AP attacks. This limitation was mitigated by using the lab mobile evil twin AP in the field evaluation. Similar approach was used by Hossen & Wenyuan’s study and the remainder of the client-side evil twin attack detection studies referenced in this dissertation.

The second limitation was that the client-side evil twin attack detection system built as part of this study is not applicable under the scenario that the attacker performs an evil twin attack using the legitimate AP’s Internet access. The author proposed that combining the detection method with other methods that were used to detect evil twin attacks using the legitimate AP’s Internet access, such as the ones referenced in this dissertation, will provide a complete evil twin attack detection system.

The study was delimited to only a hotel public Wi-Fi hotspot located in Ecuador who provide free open public Wi-Fi in hotel public areas. The conclusions reached could be extrapolated to other public Wi-Fi hotspots, as long as the design assumptions documented in this study apply. Generalization to other free open public Wi-Fi hotspots may not be warranted. Another delimitation is that the client-side evil twin attack detection system was built and evaluated using a laptop platform with Linux operating system. Generalization to other mobile platforms and operating systems may not be warranted.

## Definition of Terms

*Accuracy*: indicates how accurately the system detects evil twin AP attacks (Hossen & Wenyuan, 2014).

*Artifact*: anything that humans have created that has value to accomplish a definite function (Chandrasekaran, 1990).

*Basic service set:* a combination of an access point and one or more wireless devices (NIST, 2008).

*Eavesdropping*: an attacker monitors wireless data transmissions between devices for message content, such as authentication credentials or passwords (NIST, 2008).

*Extended service set:* a multi-BSS network (NIST, 2008).

*Evil twin attack:* an evil twin attack in a wireless local area network is a reference to a hard- or software-based 802.11 rogue Wi-Fi access point that looks like a legitimate one offered on the premises, but actually has been set up by a hacker to eavesdrop all wireless communications done by the victims (Song et al., 2010).

*Precision*: the fraction of positively detected attacks to all positively detected attacks (correctly or incorrectly) (Hossen & Wenyuan, 2014).

*Prototype system:* pilot system that is assembled, analyzed, refined, and reproduced before implementation in a production environment (Beck & Weber, 2013).

*Public Wi-Fi hotspots*: locations that provide open and free Wi-Fi internet access (Hossen & Wenyuan, 2014).

*Recall*: the fraction of positively detected attacks to all attacks that should be positively detected (Hossen & Wenyuan, 2014).

*Social lobby*:  hotel areas open to the public that provide amenities and services like free Wi-Fi, comfortable chairs, waiter service, restaurant, a bar, and coffee shop (Kelley, 2012).

*Wi-Fi*:  a trademark of the Wi-Fi Alliance and the brand name for products using the IEEE 802.11 family of standards (Singh et al., 2014).

*Wi-Fi network:* network employing the IEEE 802.11 family of standards for creating WLAN with internet facility (Singh et al., 2014).

*Wi-Fi devices:*  devices used in the Wi-Fi network (Singh et al., 2014).

*Wireless local area network:*  a group of wireless networking nodes within a limited geographic area, such as an office building or campus, that are capable of radio communication (NIST, 2008).

## Abbreviations

| | |
|---|---|
| AP | Access Point |
| BSS | Basic Service Set |
| BSSID | Basic Service Set ID |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| DS | Distribution System |
| DSRM | Design Science Research Methodology |
| DSR | Design Science Research |
| ESS | Extended Service Set |
| ETA | Evil Twin Attack |

| | |
|---|---|
| FN | False Negative |
| FP | False Positive |
| HTTPS | Hyper Text Transfer Protocol Secure |
| IP | Internet Protocol |
| IS | Information System(s) |
| ISP | Internet Service Provider |
| IT | Information Technology |
| LAN | Local Area Network |
| MAC | Media Access Control |
| NIC | Network Interface Card |
| RSSI | Received Signal Strength Identifier |
| SSID | Service Set Identifier |
| TN | True Negative |
| TP | True Positive |
| URL | Uniform Resource Locator |
| WLAN | Wireless Local Area Network |

## Summary

Chapter one of the dissertation report outlined the background for the study incorporating the problem statement which describes the need for client-side detection solutions for wireless users to be able to protect themselves from mobile evil twin attacks while using free open public Wi-Fi and the goal to develop a more effective, efficient, and practical client-side detection system linking the gap between the need for wireless security in free open public Wi-Fi hotspots and limitations in existing client-side evil twin attack detection solutions. The research questions determined the specific objectives the dissertation report focused on and were instrumental in forming the conceptual framework for the study. The first chapter also shaped the relevance and significance of the dissertation report and barriers and issues that were tackled to effectively complete the study. Finally, assumptions, limitations, and delimitations that have an overall impact on the dissertation report were presented, key terms defined and abbreviations listed.

# Chapter 2

# Brief Review of the Literature

## Overview of Topics

As groundwork to support the problem statement and research questions in this dissertation report, this section presents a review of the literature, analyzed, synthesized and organized into four main topics. The literature review focuses on justification of literature, identification of existing studies, strengths and weaknesses, gaps in literature, research methods in similar studies, and synthesis of the literature all related to the four main topics of (a) wireless security; (b) need for security in free open public Wi-Fi hotspots; (c) client-side evil twin attack detection solutions; and (d) design science research principles and methodology. The overall goal of the literature review was to guide the development of a client-side evil twin attack detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi hotspots. The design science research literature helped with the creation of DSR principles, procedures and specifications that supported the artifact construction, implementation and evaluation of the client-side evil twin attack detection system that is central to the study and has the potential to protect Wi-Fi users from mobile evil twin attacks in free open public Wi-Fi hotspots.

## Justification of Literature

The literature was selected mainly based on relevancy to design science research, client-side evil twin attack detection systems, need for security in free open public Wi-Fi hotspots, and wireless security in general. In order to support the quality of the literature review, scholarly and industry publications such as journal articles, textbooks, conference proceedings, technical reports, research reports, and online newspapers, related to the problem statement and research questions were included.  Several selected studies were also required to include observed evidence so that the researcher of this study could have support that it has been accepted by the academic community.  Most of the papers chosen in this literature review were published no earlier than 2009 because references authored before that would in all likelihood not be relevant to the industry and ongoing academic practices.

## Identification of Existing Studies

There are a number of existing studies that address the four main topics discussed in this literature review: (a) wireless security; (b) need for security in free open public Wi-Fi hotspots; (c) client-side evil twin attack detection solutions; and (d) design science research principles and methodology. This section will begin with a summary of the studies in each of the four main topic areas and will help provide the groundwork for synthesis of the literature at the end of the section.

## Wireless Security

This section presents a review of the literature relevant to wireless local area network (WLAN) security in general.  This section begins with an introduction to WLAN, the basic WLAN components, and architecture of WLAN.  Subsequently, it describes various security threats of WLAN, standards for WLAN security, and concludes with several practical solutions for securing WLAN. The studies chosen for this section offer an introductory backdrop on the topic of wireless security.

### *Introduction to WLAN*

Wireless local area network (WLAN) is a group of wireless networking nodes within a limited geographic area, such as an office building or campus, that are capable of radio communication. In 1997, IEEE first approved the IEEE 802.11 international interoperability standard for WLANs. In 1999, IEEE ratified two amendments to the IEEE 802.11 standard, IEEE 802.11a and IEEE 802.11b, that define radio transmission methods and modulation techniques. WLAN equipment based on IEEE 802.11b rapidly became the leading wireless technology. IEEE 802.11b equipment transmits in the 2.4 GHz band, offering data rates of up to 11 Mbps. IEEE 802.11b was proposed to deliver performance, throughput, and security features comparable to wired LANs. IEEE 802.11a operates in the 5 GHz Unlicensed National Information Infrastructure (UNII) frequency band, delivering data rates up to 54 Mbps. In 2003, IEEE announced the IEEE 802.11g amendment, which details a radio transmission method that also operates in the 2.4 GHz ISM band and can sustain data rates of up to 54 Mbps. Furthermore, IEEE 802.11g-compliant products are backward compatible with IEEE 802.11b-compliant products (NIST, 2008).

In 2006, the first IEEE 802.11n draft was offered to enhance the range and speed of WLANs up to theoretical speeds of 300 Mbps. IEEE 802.11n maintains backward compatibility with IEEE 802.11a/b/g WLANs because it runs on both the 2.4 GHz ISM band and the 5.0 GHz UNII band. Throughput is enhanced over its predecessors by exploiting wider bandwidth channels and devices supplied with multiple antennas to better tap into RF signal. Moreover, IEEE 802.11n almost doubles the effective range of the WLAN (NIST, 2008). In 2014, IEEE approved IEEE 802.11ac which is planned to achieve higher multi-user throughput in wireless local area networks (WLANs). IEEE 802.11ac is intended to enhance WLAN user experience by offering data rates up to 7 Gbps in the 5 GHz band, more than 10 times the speed that was previously standardized (Kelly, 2014).

The network employing the IEEE 802.11 family of standards for creating WLAN with Internet facility is called Wi-Fi network, and the devices operating in that network are called Wi-Fi devices. Wi-Fi is a trademark of the Wi-Fi Alliance and the brand name for products using the IEEE 802.11 family of standards. The advantages of Wi-Fi networks comprise: convenience, mobility, productivity, deployment, expandability and cost. The disadvantages of using Wi-Fi networks are: security, range, reliability, and speed (Kirankumar, Babu, Prasad, and Wishnumurthy, 2012; Singh, Mishra, and Barwal, 2014).

WLAN technology generates new threats. For example, since communications take place "through the air" riding on radio frequencies, the risk of interception is greater than with wired networks. If the message is not encrypted, or encrypted with a weak algorithm, the attacker can read it, thereby conceding confidentiality. Data encryption is the principal means of security in a WLAN. Without encryption, any ordinary wireless device can read all traffic in a network, and in 802.11 WLANs, encryption is optional. The overarching security goals for WLAN are identical to those of wired networks: preserving confidentiality, ensuring integrity, and maintaining availability of the information and information systems (Habibi, Seyed, and Samadi, 2009; Kirankumar et al., 2012).

### *WLAN Components*

One important advantage of WLAN is the ease of its installation. WLAN systems can remove the requirements of pulling cable through walls and ceilings. The network architecture of a WLAN is very basic. Key components of a WLAN are access points (APs) and network interface cards (NICs) (Feng, 2012; Singh et al., 2014).

Access Point (AP) is the wireless equivalent of a LAN hub. An AP is usually connected to the Ethernet architecture through an Ethernet port. The AP includes a radio and antenna for communication with client devices. Access points function within a particular frequency spectrum and use 802.11 modulation techniques specified in the standard. It also informs the wireless clients of its availability, and authenticates and associates wireless clients to the wireless network (Feng, 2012; Singh et al., 2014).

Wireless NICs connect wireless devices such as laptop computers, PDAs, mobile telephones, and other consumer electronic devices to a wireless network either in ad-hoc peer-to-peer mode or in infrastructure mode with APs. NICs scan the specified spectrum for potential connectivity and associate to an AP or another wireless device (Feng, 2012; Singh et al., 2014).

### *WLAN Architecture*

The IEEE 802.11 standard outlines two basic WLAN topologies: ad-hoc network and infrastructure network. An ad-hoc network is a peer-to-peer network between wireless clients, and no APs are part of the architecture. An infrastructure network consists of APs connected to a distribution system (DS), usually a wired network, and wireless clients. Infrastructure is the most frequently used mode for WLANs (NIST, 2008).

### *Ad-hoc Network*

This mode of operation occurs when two or more wireless devices communicate directly to each other. This is called ad-hoc Wi-Fi transmission. The name ad-hoc is used because the network is set up typically for express purpose and for a short time. One of the key advantages of ad-hoc WLANs is that theoretically they can be established anytime and anywhere, permitting multiple users to create wireless connections cheaply, quickly, and easily with minimal hardware and user maintenance. Ad-hoc networks have no connection to the other networks. A set of wireless devices configured in this ad-hoc manner is known as an independent basic service set (IBSS). Figure 2 represents a sample IBSS that includes a mobile telephone, laptop computer, and a PDA interconnecting via IEEE 802.11 technology. The circle in Figure 2 illustrates the signal range of the devices, which is imperative to consider because this limits the coverage area within which the stations can continue in communication (NIST, 2008).

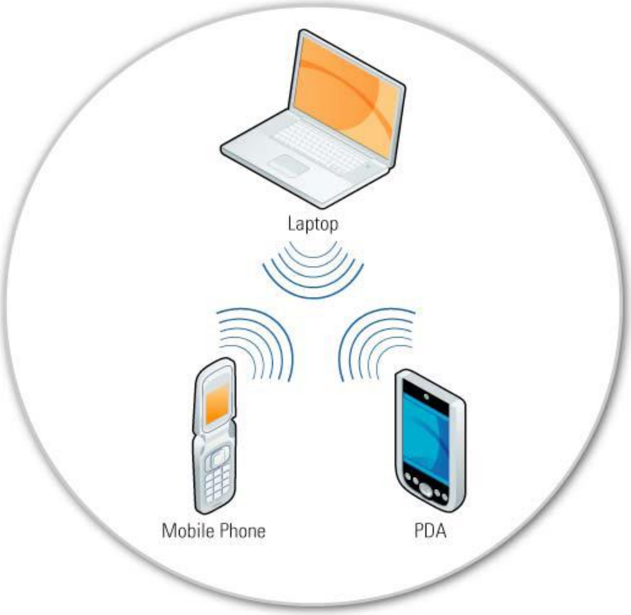


*Figure 2.* Ad-hoc network.

### *Infrastructure Network*

An infrastructure network involves wireless devices and access points. In an infrastructure network, the wireless clients connect with each other by having an access point. An access point is the device that operates as a bridge from the wireless network to the wired network. When access points connect to a distribution system (such as Ethernet), they support the creation of multiple coverage cells that enable roaming throughout a facility. A combination of an AP and one or more wireless devices is called Basic Service Set (BSS). The use of multiple APs connected to a single distribution system (DS) allows for the creation of wireless networks of arbitrary size and complexity. In the IEEE 802.11 specification, a multi-BSS network is referred to as an extended service set (ESS). Figure 3 conceptually depicts a network with both wired and wireless capabilities, comparable to the architecture of a public Wi-Fi environment. It displays two APs with corresponding BSSs, which comprise an ESS. The ESS is joined to the wired enterprise network or DS, which, in turn, is linked to the Internet. This architecture could permit various wireless devices, such as laptop computers and PDAs, to access network resources and the Internet. Also, the use of an ESS affords the opportunity for IEEE 802.11 WLAN devices to roam between APs while maintaining network connectivity (NIST, 2008). Public Wi-Fi hotspots usually have multiple wireless access points and share the same SSID. This allows public Wi-Fi users to move around the public spaces with their mobile devices without being disconnected from the network. While moving around the public spaces, the Wi-Fi user will disassociate and associate to the access point with the best signal strength. All of this is transparent to the user.

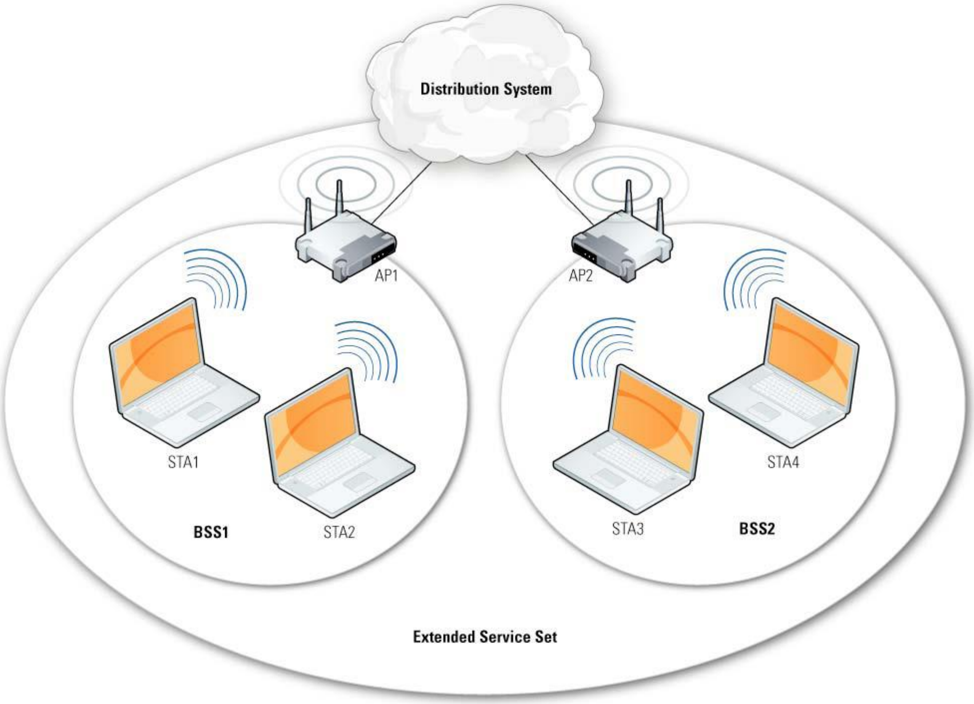


*Figure 3.* Infrastructure network.

*Security Threats of WLAN*

Generally, wireless networks are more susceptible to security attacks than wired networks, attributable to the broadcast nature of the transmission. Despite the productivity, convenience and cost advantage that WLAN presents, the radio waves used in wireless networks generate a risk that the network can be hacked. Most threats against wireless networks include an attacker with access to the radio link between wireless devices (Kirankumar et al., 2012; Singh et al., 2014).

According to NIST (2008), WLAN technologies usually must support several security objectives. The most common security objectives for WLANs are:

1. *Confidentiality*: Ensure that communication cannot be read by unauthorized parties.
2. *Integrity:* Detect any intentional or unintentional changes to data that occur in transit.

3. *Availability*: Ensure that devices and individuals can access a WLAN and its resources whenever needed.

NIST (2008) indicated that network security attacks against WLANs are usually divided into passive and active attacks. These two broad classes are then subdivided into other types of attacks. All are defined below:

1. *Passive Attack:* An attack in which an unauthorized individual acquires access to an asset and does not modify its content or actively attack or disrupt a WLAN. There are two types of passive attacks:
   - *Eavesdropping*: The attacker monitors wireless data transmissions between devices for message content, such as authentication credentials or passwords. An example of this attack is an intruder monitoring transmissions on a WLAN between an AP and a connected device.
   - *Traffic analysis:* The attacker gains intelligence by monitoring the transmissions for patterns of communication. A substantial amount of data is contained in the flow of messages between communicating parties. This method is subtler than eavesdropping.
2. *Active Attack:* an attack whereby an unauthorized party makes modifications to a message, data stream, or file. It is feasible to detect this type of attack, but it may not be avertible. Active attacks consist of four types (or a combination thereof):
   - *Masquerading*: The attacker impersonates an authorized user to gain access to certain unauthorized privileges.
   - *Replay*: The attacker monitors transmissions (passive attack) and retransmits messages posing as the legitimate user.

- *Message modification*: The attacker alters a legitimate message by deleting, adding to, changing, or reordering the message.
- *DoS*: The attacker prevents or prohibits the normal use or management of a WLAN.

### *Standards for WLAN Security*

This section describes the security features provided by IEEE 802.11 WLAN standards.

### *Wired Equivalent Privacy (WEP)*

Wired Equivalent Privacy (WEP) is a standard encryption for wireless networking leveraging the Rivest Cipher 4 (RC4) algorithm with two sides of a data communication. It is a user authentication and data encryption system from IEEE 802.11 applied to defeat security threats. Essentially, WEP offers security to a WLAN by encrypting the information transmitted over the air, so that only the receivers who have the correct encryption key can decrypt the information (Habibi et al., 2009; Singh et al., 2014). WEP was originated by the IEEE to deliver the following three basic security services: (a) authentication to verify the identity of communicating client stations; (b) confidentiality to use encryption to offer wireless networks with the same or similar privacy achieved by an unencrypted wired network; and (c) integrity to ensure that messages were not modified in transit between wireless clients and APs. IEEE 802.11 configurations that rely on WEP have several well-documented security problems. The IEEE and the Wi-Fi Alliance acknowledged the scope of the problems and developed short-term and long-term strategies for rectifying the situation. In early 2003, the Wi-Fi

Alliance, in coordination with the IEEE 802.11 Working Group, developed the Wi-Fi Protected Access (WPA) security enhancement to replace WEP (NIST, 2008).

### *Wi-Fi Protected Access (WPA)*

WPA was released as a temporary measure until a robust IEEE 802.11 security standard could be developed and approved. WPA includes two main characteristics: IEEE 802.1x and the Temporal Key Integrity Protocol (TKIP). The IEEE 802.1x port-based access control provides a framework to allow the use of robust upper-layer authentication protocols. It also enables the use of session keys that allow the rotation of cryptographic keys. TKIP contains four new features to enhance the security of IEEE 802.11: TKIP extends the IV space, allows for per-packet key construction, provides cryptographic integrity, and provides key derivation and distribution. Furthermore, it addresses the critical need to periodically change encryption keys. However, WPA has significant flaws and does not provide the level of security that Wi-Fi Protected Access II (WPA2)/802.11i can (NIST, 2008).

### *WPA2/ 802.11i*

Habibi et al. (2009) stated that the WPA2 and 802.11i terms are often used interchangeably. According to Habibi et al. (2009), Feng (2012), and Singh et al. (2014), the WPA2 standard specifies two modes of security:

1. In “personal” mode a pre-shared secret is used, much like WEP or WAP. Access points and clients are all manually configured to use the same secret of up to 64 ASCII characters, such as “this_is_our_secret_password.” An actual 256-bit

randomly generated number may also be used, but this is difficult to enter manually into client configurations.

2. The “enterprise” security is based on 802.1x, the EAP authentication framework, and secure key distribution. 802.1x was originally designed for wired Ethernet networks. The following discussion of 802.1x is divided into three separate sections: Point-to-Point Protocol (PPP), Extensible Authentication Protocol (EAP) and 802.1x itself.

### *PPP*

The Point-to-Point Protocol (PPP) originally emerged as an encapsulation protocol for transporting IP traffic over point-to point links. PPP also launched a standard for the assignment and management of IP addresses, asynchronous (start/stop) and bit-oriented synchronous encapsulation, network protocol multiplexing, link configuration, link quality testing, and error detection. By any standard, PPP is a good protocol. However, as PPP usage grew, hackers quickly uncovered its limitation in terms of security. This leads to the designation of a new authentication protocol, called Extensible Authentication Protocol (EAP).

### *EAP*

The Extensible Authentication Protocol (EAP) is a general authentication protocol defined in IETF (Internet Engineering Task Force) standards. It was originally designed for use with PPP. It is an authentication protocol that presents a generalized framework for several authentication mechanisms. These consist of Kerberos, public key, smart cards and one-time passwords. With a standardized EAP, interoperability and compatibility across authentication methods become simpler. For instance, when a user dials a remote access server (RAS) and use EAP as part of the PPP connection, the RAS does not need to know any of the details about the authentication system. Only the user and the authentication server have to be synchronized. By supporting EAP authentication, RAS server does not actively participate in the authentication dialog. Instead, RAS just re-packages EAP packets to handoff to a remote access dial in user service (RADIUS) server to make the actual authentication decision.

### *802.1x*

IEEE 802.1x relates to EAP in a way that it is a standard for carrying EAP over a wired LAN or WLAN. There are four important entities that expound upon this standard:

1. *Authenticator:* Authenticator is the entity that requires the entity on the other end of the link to be authenticated. An example is wireless access points.
2. *Supplicant*: Supplicant is the entity being authenticated by the authenticator and desiring access to the services of the authenticator.
3. *Port Access Entity (PAE):* It is the protocol entity associated with a port. It may support the functionality of authenticator, supplicant or both.
4. *Authentication Server:* Authentication server is an entity that provides authentication service to the authenticator. It may be co-located with authenticator, but it is most likely an external server. It is typically a RADIUS server.

### *Practical Solutions for Securing WLAN*

This section presents the use of hardware and software solutions to help secure the WLAN environment.

### *VPN*

An alternate method of realizing confidentiality and integrity protection is using a virtual private network (VPN). A VPN is a virtual network, built on top of existing physical networks, that can afford a secure communications mechanism for data and IP information transmitted between networks. VPNs are often used to enable the secure transfer of sensitive data across public networks, such as the Internet, for remote access, telework, and other situations encompassing connecting multiple locations. VPNs can also be set up within a single network, such as a WLAN, to safeguard sensitive communications from other parties on the network. A variety of VPN technologies exist, such as Internet Protocol Security (IPsec) VPNs and Secure Sockets Layer (SSL) VPNs. One way to use VPNs to protect WLAN communications is to establish a VPN tunnel between the WLAN client device and a VPN concentrator that is behind the AP. With an IPsec VPN, security services are provided at the network layer of the protocol stack, which will secure all applications and protocols operating at layer 3 and above. The VPN security services are independent of layer 2 wireless security and are recommended to be used if the underlying wireless security mechanisms are weak (NIST, 2005).

### *Universal Authentication Mechanism (UAM)*

With UAM, any device is permitted to associate to the Wi-Fi access point and is allotted an IP address and other network information such as the standard gateway automatically via DHCP. After association, the user opens a web browser and enters any URL. A transparent HTTP proxy (also called captive portal) on the AP (or the underlying infrastructure) captures the request and redirects it to a login page. In the case of free open public Wi-Fi, the user is usually just required to accept the terms of use.  Now the user’s primary HTTP request is sent and the response is delivered to the user (Szongott et al., 2012).

## Need for Security in Free Open Public Wi-Fi Hotspots

The review of the literature in this section will attempt to discover the need for security in free open public Wi-Fi hotspots. The studies included in this section were chosen because they demonstrate a significant trend toward the need for wireless security in free open public Wi-Fi hotspots.

In order to build the foundation supporting the need for wireless security in free open public Wi-Fi, Han et al. (2009, 2011), Song et al. (2010, 2012), and Kim et al. (2012) indicated in their studies that the Internet has become a part of our everyday life and the use of IEEE 802.11 based wireless local area networks, WLANs, or Wi-Fi has rapidly increased in popularity in recent years for accessing the Internet. In recent years, Internet usage shifted from stationary to mobile devices such as laptops, tablets, or smartphones with a wireless connection to the network (Lanze et al., 2014). Wi-Fi market reached 6.4 billion in 2011 and a rapid growth is forecasted in the upcoming years as most of mobile devices (e.g., laptops, tablets, smartphones, etc.) have Wi-Fi capability (Myslewski, 2011; IDC, 2017).

Wi-Fi’s popularity is due to the following reasons: mobility, flexibility, scalability, and ease of installation (Nikbakhsh et al., 2012). Although mobile cellular networks (e. g., 3G/4G LTE) have gained an increasing influence, the importance of Wi-Fi networks remains crucial. Wi-Fi networks provide faster connectivity, offer unmetered service whenever mobile networks are unavailable, overloaded, or overpriced (e.g., in roaming) and are indispensable for devices that do not have hardware to access mobile cellular networks, e. g., laptops or many tablets (Lanze et al., 2014).

Both users and providers benefit significantly from public Wi-Fi hotspots. Users receive wireless Internet access and providers attract new potential clients (Monica & Ribeiro, 2011). Many public avenues have set up Wi-Fi access points to provide free wireless Internet service in order to attract and better serve their customers (Hossen & Wenyuan, 2014; Lanze et al., 2014; Nakhila et al., 2015).

Han et al. (2009, 2011) indicated that as users' expectations of wireless availability increases, the security of such networks becomes even more important. Additionally, Kim et al. (2012) and Nikbakhsh et al. (2012) stated in their studies that the growing popularity of WLANs increases the risk of wireless security attacks. Furthermore, there is a negative incentive for providers to implement security mechanisms, because the goal of the hotspots is to provide convenience and to attract customers. For instance, public Wi-Fi hotspots provide free, open, and zero liability Internet access to customers (Hossen & Wenyuan, 2014). Nakhila et al. (2015) indicated that clients will only need to search the airwave and connect to the wireless network. No means of encryption or authentication is used besides the wireless network name. Wi-Fi's popularity makes it an attractive target for intruders to compromise and to eavesdrop wireless client information (Nakhila et al., 2015).

According to Song et al. (2010, 2012), Hossen & Wenyuan (2014), and Nakhila et al. (2015), while users can access Wi-Fi Internet hotspot connections in public more easily, they become more vulnerable to fraud and identity theft, referred to as evil twin attacks (ETA). This is a threat that can severely compromise the security of wireless users.

Moreover, lack of knowledge and awareness possessed by the user make this issue extremely disturbing (Nikbakhsh et al., 2012). Due to its severity, the evil twin attack has gained a significant amount of interest in the media and research community (Han et al., 2009, 2011; Song et al., 2010, 2012; Lanze et al., 2014).

### *Evil Twin Attack*

An evil twin attack in a wireless LAN refers to a hard- or software-based 802.11 rogue Wi-Fi access point (AP) that appears to be a legitimate one offered on the premises, but actually has been set up by a hacker to “eavesdrop” on all wireless communications done by the victims (Song et al., 2010, 2012; Monica & Ribeiro, 2011; Lanze et al., 2014; Hossen & Wenyuan, 2014; Nakhila et al., 2015; Hsu et al., 2015). An evil twin AP mimics the identity of a legitimate AP by cloning its characteristics, such as SSID, MAC, or IP address, to be able to trap users to hijack their Internet connection for monetary gain (Nikbakhsh et al., 2012; Kim et al., 2012; Lanze et al., 2014). In the existing literature, the terms evil twin AP, rogue AP, fake AP and spoofed AP are used synonymously.

### *Attack Scenarios*

Song et al. (2010, 2012), Nikbakhsh et al. (2012), Hossen & Wenyuan (2014), and Nakhila et al. (2015) discovered the following attack scenarios:

1. *Using the legitimate AP’s Internet access (Multihop attack):* The attacker connects his device to a legitimate AP for accessing the Internet. In this scenario, the evil twin AP can itself behave as a normal Wi-Fi client and uses the legitimate AP to connect with the Internet. All the wireless traffic from the victim will pass through the attacker’s node. In the literature, this type of attack scenario is denoted as multihop attacks. Figure 4 illustrates the multihop attack scenario.

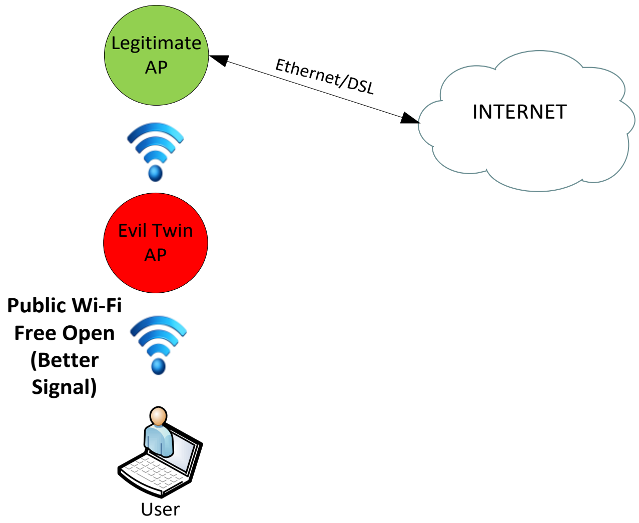


*Figure 4.* Multihop attack scenario.

2. *Using mobile Internet access (Mobile attack):* The attacker uses mobile Internet, e.g. 3G/4G LTE, as the access network for connecting to the Internet. A hotspot router can act as an evil twin AP. Also, a smartphone with mobile AP functionality built in operating systems such as Android or iOS, can act as an evil twin AP and thus can allow Wi-Fi clients to use mobile Internet service of the smartphone. In the literature, this type of attack scenario is denoted as mobile attacks. Figure 5 illustrates the mobile attack scenario.

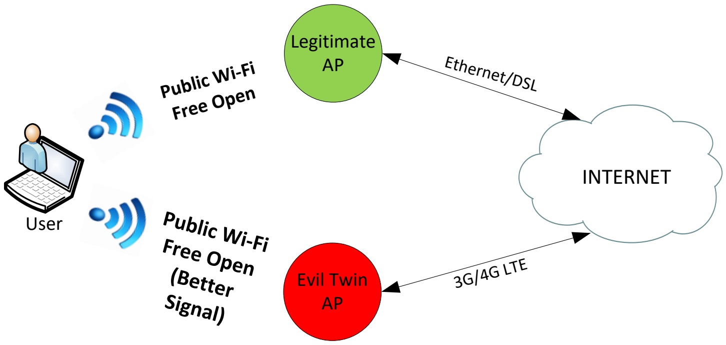


*Figure 5.* Mobile attack scenario.

### *Evil Twin AP Set Up*

It is easy for an adversary to create an evil twin AP in a public Wi-Fi hotspot using a Wi-Fi enabled device, e.g., laptop, smartphone, etc. (Song et al., 2010, 2012; Lanze et al., 2014). By using a free, fully-automated software (e.g. aircrack-ng), an attacker can simply configure a Wi-Fi enabled device to be an evil twin AP to mimic the legitimate access point used in a free public Wi-Fi area (Song et al., 2010, 2012; Hossen & Wenyuan, 2014; Lanze et al., 2014). Additionally, all common mobile operating systems including Android and iOS are capable of creating a wireless AP using mobile hotspot functionality. Hence, this process can be performed directly from smartphones, without attracting the attention of anybody in the vicinity (Lanze et al., 2014).

### *Launch of Evil Twin Attacks*

An evil twin attack is easy to launch at public places due to the lack of security mechanisms. Han et al. (2009, 2011), Song et al. (2010, 2012), Monica & Ribeiro (2011), and Hsu et al. (2015) stated in their studies that there are three strategies for attackers to attract victims into connecting to their rogue access points. The first strategy is by having a rogue AP with the SSID of the targeted public Wi-Fi network physically set closer to clients so that its signal can be stronger than the legitimate access points. The attacker can also intensify the transmission power of the evil twin AP. This strategy works, since several operating systems choose the AP with the strongest signal strength when several APs with the same SSID are available, as these operating systems believe different APs with the same SSID belong to the same hotspot. Also, wireless users tend to choose the network with the highest signal strength when manually selecting a network to connect to. The wireless users basically assume that all the APs are legitimate. The second strategy uses the automatic re-association feature that several end-user systems provide. These systems have preferred network lists, containing the SSID names of the networks a user has connected to in the past. The attacker simply choses the evil twin AP SSID name as the most commonly used SSID names, and waits for victims to connect. Finally, the third strategy involves using a denial-of-service attack against 802.11 networks. The rogue AP can passively wait for users to connect to it, or actively send fake de-associate frames to force users to change connections. The loss of connectivity caused from the continuous disassociations, forces wireless users to choose other available wireless networks.

Song et al. (2010, 2012) and Monica & Ribeiro (2011) indicated that evil twin attacks can be hard to trace. The attacker can shut off the attacks suddenly or randomly after accomplishing the malicious goals. In a very short time frame, the attacker may already have compromised public Wi-Fi user’s sensitive information, such as passwords or credit card information. According to Hsu et al. (2015), when a user connects to an evil AP, it is exposed under an open connection to the attacker causing privacy data leakage. Detecting evil twin access points is the first step in dealing with this problem.

In addition, Song et al. (2010, 2012), Monica & Ribeiro (2011), Choi, Chang, Ko, and Hu (2011), Cheng, Wang, Cheng, Mohapatra, and Seneviratne (2013), Lanze et al. (2014), and Hossen & Wenyuan (2014) found that WPA2, VPN and UAM solutions are not appropriate for protecting against evil twin attacks. WPA2 is a mechanism that has to be configured and carefully maintained by an operator, and operators of public Wi-Fi hotspots in particular have no incentive to provide such a service. Additionally, VPN technology is not easily accessible for all users since security service providers usually charge a monthly fee. Finally, with UAM, the initial URL accessed by the user is redirected to a captive portal page to only accept terms of use of free Wi-Fi. This page can be easily emulated by an attacker. In addition, UAM at hotspots does not allow the user to confirm the integrity of the hotspot or its provider. Hence, this method does not offer any security at all for the user pertaining to the evil twin attack.

## Client-side Evil Twin Attack Detection Solutions

The client-side evil twin detection studies selected and summarized in this section provide insight on limitations of existing client-side detection solutions and also direction related to the design, construction, deployment, and evaluation of a client-side detection artifact that can be used to detect an evil twin attack in free open public Wi-Fi environments. All the studies found related to this domain focus on the construction of a client-side detection artifact based on best practices and industry standards but not on design principles derived from DSR since there are no studies that effectively address this innovative method of artifact construction, implementation and evaluation.

In an early study, Han et al. (2009) developed a client-side timing-base method for detection of rogue access points based on round trip time (RTT) calculation between the wireless user and the DNS server, and does not require network administrator assistance. Their RTT-based method helps distinguish the route through a rogue AP from that through a legitimate AP (one hop versus two-hop wireless channels). Han et al. (2009) found that this additional hop introduces an unavoidable time delay. In a later study, Han et al. (2011) extended their work by using an outlier algorithm to reduce false detection, and dynamically adjusting the number of samples in each test to reduce detection time without sacrificing accuracy.  Their method requires knowledge of the wireless hotspot network infrastructure.  Also, their method is based on the attacker using the legitimate wireless network gateway to pass through client data traffic, assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP.  Their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open public Wi-Fi networks.  Their method assumes that the user has not connected to the target public Wi-Fi network in the past.  Han et al. (2009, 2011) evaluated their solution in the lab and field using their own evil twin AP.  Their study set a benchmark for creating client-side detection methods that allow wireless users to use their station to independently detect whether an AP is legitimate or not without additional equipment and the assistance of a wireless network administrator in free open public Wi-Fi hotspots.

Additionally, using timing measurements and also based on the evil twin AP utilizing the legitimate AP to connect to the Internet, Song et al. (2010) introduced a prototype system called “ETSniffer” (Evil Twin Sniffer) based on Interpacket Arrival Time (IAT) to detect evil twin access points by distinguishing a one-hop from a two-hop wireless network setting between the wireless client and the remote IAT server (custom server). Their method does not require administrator assistance.  Two methods were presented as part of this study.  The first method is called Trained Mean Matching (TMM) and requires knowing the distribution of server IAT as trained knowledge and the second method is called Hop Differentiating Technique (HDT) and does not have such a requirement.  Their study suggested that HDT improves TMM by removing the training requirement.  Both algorithms utilize the wireless IAT network statistic, consider the influencing factors of received signal strengths (RSSs) and wireless network saturation, and employ Sequential Probability Ratio Test (SPRT) technique to make the final detection.  As an improvement, Song et al. (2012) provided additional options for IAT remote servers that can be utilized to measure IAT statistics.  Their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open public Wi-Fi networks.  Their method assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP.  Their method assumes that the user has not connected to the target public Wi-Fi network in the past.  Song et al. (2010, 2012) evaluated their solution in the lab and field using their own evil twin AP.  Their work made an important contribution by proposing the first client-side evil twin attack detection solution that did not require prior knowledge of the wireless hotspot network infrastructure and network administrator assistance.

As the technology continued to move toward client-side evil twin detection systems, Monica & Ribeiro (2011) developed a detection solution called “WiFiHop”.  Their method does not require network administrator assistance.  This detection system is based on the behavior of the legitimate AP without depending on timing to detect a multihop setting between the wireless user and the Internet.  Their solution requires the implementation of a script (echo server) on a public hosting server.  Their method is based on the evil twin AP relaying traffic to the Internet using the legitimate AP and is technology independent.  Monica & Ribeiro found that when an evil twin attack is in place, the user's data must transit the wireless channel between the evil twin and the legitimate AP. If an extra wireless hop is detected, then the presence of an evil twin AP is confirmed.  Their method was based on the attacker using the legitimate wireless network gateway to pass through client data traffic. Their system is automated with no intervention from users.  Their method works with any type of IEEE 802.11 based wireless networks and Wi-Fi enabled devices.  Furthermore, their method assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP.  Their method assumes that the user has not connected to the target public Wi-Fi network in the past.  Monica & Ribeiro (2011) evaluated their solution in the lab and field using their own evil twin AP. Their major contribution was to develop a solution that does not depend on timing to detect a multihop setting, does not require network administrator assistance, and that it operates in both free open and encrypted public Wi-Fi networks.

In support of warning users to avoid connecting to evil twin access points in public Wi-Fi hotspots, Nikbakhsh et al. (2012) developed a client-side approach based on traceroute that compares the gateways and routes that a packet travels to determine whether an access point is legitimate or not without the assistance from a wireless LAN operator.  If the legitimate AP and evil twin AP have the same IP addresses with different trace route (IP spoofing), their method does not have any references to check which one is the authorized access point, therefore it just warns the user about an evil twin attack. Their method was based on the attacker using the legitimate wireless network gateway to pass through client data traffic. Their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open public Wi-Fi networks. Furthermore, their method assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP. Their method assumes that the user has not connected to the target public Wi-Fi network in the past. Their approach was not implemented or evaluated in a lab environment or in the field.

During the same time, Kim et al. (2012) developed a client-side evil twin attack detection method for smartphones only based on received signal strengths (RSSs), an online detection algorithm, and does not require network administrator assistance. Their method measured RSSs from both the legitimate and evil twin access points on the smartphone and used normalization of collected signal strengths for accurate measurement. Finally, the method classified signal strengths that are highly correlated to others based on a defined threshold value. Highly correlated RSSs are considered fake signals from an evil twin access point. Kim et al. (2012) made the assumption that the attacker was using the legitimate wireless network gateway to pass through client data traffic. Their system is automated with no intervention from users. Their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open public Wi-Fi networks. Additionally, their method assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP. Their method assumes that the user has not connected to the target public Wi-Fi network in the past. Kim et al. (2012)

evaluated their solution in the lab and field using their own evil twin AP. Their work made an important contribution by proposing the first client-side evil twin detection solution for a smartphone that works on open and encrypted networks.

A recent study conducted by Lanze et al. (2014) addresses the problem of lack of client-side evil twin attack detection solutions for public Wi-Fi hotspot users to independently verify whether an access point is legitimate or not through a method for detection of software-based evil twin attacks (e.g. aircrack-ng) and without network administrator assistance. Their method separates software access points from legitimate hardware access points. Lanze et al. (2014) found that when software emulates hardware behavior, it presents a significant timing inaccuracy due to processing delays and leaks information that can be used for detection. Further, their method explains why airbase-ng fails to imitate a hardware AP in regards to the accuracy of Timing Synchronization Function (TSF) timestamps in beacon frames. Their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open public Wi-Fi networks. In addition, their method assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP. Their method assumes that the user has not connected to the target public Wi-Fi network in the past. Their solution was only implemented in a lab environment. Lanze et al. (2014) used their own evil twin AP on the evaluation.

Similarly, using the legitimate AP to connect to the Internet, Hsu et al. (2015) proposed a client-side evil twin attack detection system called “ET Detector” based on redirection behavior. Their method does not require administrator assistance. By operating the wireless network interface controller (WNIC) in monitor mode (which is able to capture all packets that conform to its monitoring channel and protocol) and by examining the captured packets, users can simply and accurately discover the evil twin attack. The system has two detection mechanisms: default testing and secondary device testing. Default testing only works when a user is not the only one using public Wi-Fi in a hotspot. Otherwise, the system will be forced to use secondary testing which requires an extra Wi-Fi device with no sensitive data on it to associate to the target AP to make the detection. Their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open public Wi-Fi networks. Their method assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP. Their method assumes that the user has not connected to the target public Wi-Fi network in the past. Hsu et al. (2015) evaluated their solution in the lab and field using their own evil twin AP.

Szongott et al. (2015) proposed a detection system called Mobile Evil Twin Detection System (METDS) for smartphones based on context-based recognition that can help mitigate evil twin attacks, and does not require administrator assistance. To detect evil twin APs, the algorithm of the METDS utilizes the following parameters to describe and verify an access point's environment: SSID, BSSID, cell tower information, and device’s location. Their method only works if the METDS system has previously been run in the hotspot. In this case, METDS already has an appropriate dataset that can be used to verify the current environment. Their method requires previous knowledge of the target network and also an external server to store learned data. Their system is automated with no intervention from users. Also, their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open public Wi-Fi networks. Their method assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP. Their method assumes that the user has not connected to the target public Wi-Fi network in the past. Their solution was evaluated in a lab using simulations based on real-world data.

Continuing in the domain of client-side evil twin detection, Hossen & Wenyuan (2014) introduced a method called Client end Evil Twin Access Point Detector (CETAD) to detect evil twin attacks without network administrator assistance. Their method leverages public servers and was implemented as an application in a smartphone. Their application included two detection techniques: ISP-based and timing-based. The application utilized the ISP-based detection technique, and if not successful, used the timing-based detection technique. The ISP-based technique was used to detect mobile attacks as the ISP information of a legitimate AP and an evil twin AP are different. It detects whether or not different gateways are used by multiple APs in one hotspot location that have the same SSID. Hossen & Wenyuan’s ISP-based method for mobile attacks uses a public website to gather the global IP address shared by the legitimate APs. Timing-based technique was used to detect multihop attacks because the attacker’s evil twin AP uses the legitimate AP as the gateway. Hossen & Wenyuan (2014) claim that when the evil twin attack is launched utilizing the victim’s Internet, RTT values vary significantly.  Hossen & Wenyuan assume that the BSSID of the hotspot APs are unique and use that as a reference to switch between different APs with the same SSID in the hotspot.  Hossen & Wenyuan assume that the evil twin AP is in a different subnet as the legitimate AP.  Their method assumes detection of all the APs in the public Wi-Fi network during the initial wireless network scanning and that the client is able to associate to all the APs in the public Wi-Fi network.  Their system is automated with no intervention from users.  Their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open Wi-Fi networks.  Furthermore, their method assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, and warns the wireless user of an evil twin attack before any traffic is transmitted.  Hossen & Wenyuan (2014) claim that after the attack has been detected, the application allows the wireless user to connect to the legitimate AP, but this was not included in their algorithm.  Furthermore, their method does not cover the scenarios when the attacker blocks access to the public website or when the attacker presents an invalid certificate while ISP information is retrieved from the public website.  Their method assumes that the user has not connected to the target public Wi-Fi network in the past.  Hossen & Wenyuan (2014) was evaluated in the lab and field using their own evil twin AP.  Their main contribution was evaluating the effectiveness of the client-side detection system at a large scale in uncontrolled environments.

In a similar study, Nakhila et al. (2015) also presented a client-side detection method for mobile attacks that does not require network administrator assistance and detects whether or not different gateways are used by multiple APs in one hotspot location that have the same SSID.  Their detection technique relies on an SSL/TCP connection to a remote public server, and detects the changing of wireless network gateway's public IP address in the middle of the SSL/TCP connection. Nakhila et al. (2015) assume that a mobile attack is not executed before the client establish a secure connection to the remote server. Also, Nakhila et al. assume that the BSSID of the hotspot APs are unique and use that as a reference in their method to switch between different APs with the same SSID in the hotspot. Nakhila et al.'s (2015) method only works when the evil twin AP is in the same subnet as the legitimate AP. Their method assumes the attacker uses a different gateway from a legitimate AP. If the attacker uses a legitimate gateway to pass wireless client data, the proposed detection method will not work. Their method assumes detection of all the APs in the public Wi-Fi network during the initial wireless network scanning and that the client is able to associate to all the APs. Nakhila et al.'s method does not cover the scenario when the attacker blocks access to the public website. Their system is automated with no intervention from users. Their method works with any type of IEEE 802.11 based wireless networks, Wi-Fi enabled devices, and with free open Wi-Fi networks. Furthermore, their solution assumes the attacker is connected when the wireless user connects to the public Wi-Fi hotspot, warns the wireless user of an evil twin attack before any traffic is transmitted, and after detection, it does not allow the user to connect to a legitimate AP. Their method assumes that the user has not connected to the target public Wi-Fi network in the past. Their solution was only implemented in a lab environment. Nakhila et al. (2015) used their own evil twin AP on the evaluation.

The client-side detection artifact constructed as part of this dissertation linked the gap between the need of wireless security in free open public Wi-Fi hotspots and limitations in existing client-side evil twin attack detection solutions.

Appendix A shows existing studies requirements and limitations mapping based on the literature review.

## Design Science Research Principles and Methodology

Design theory played a significant role in the development of DSR principles that were used in the effective construction, implementation, and evaluation of a client-side evil twin attack detection system to allow wireless users of free open public Wi-Fi to detect mobile evil twin attacks.

According to Walls, Widmeyer, and El Sawy (1992), IS design theories are prescriptive, which integrates normative and descriptive theories into design paths intended to produce more effective information systems. IS design theories prescribe effective development practices (methods) and a type of system solution (instantiation) for a particular class of user requirements (models). Further, Walls et al. (1992) indicated that explanatory theories tell “what is”, predictive theories tell “what will be”, normative theories tell “what should be”, and design theories tell “how to/ because”.

In support of Walls et al.’s (1992) study, March and Smith (1995) found that design science offers prescriptions and creates artifacts that embody those prescriptions. Design science attempts to create things that serve human purposes, it is technology oriented and its products (constructs, models, methods, and implementations) are assessed against criteria of utility to a community of users (e.g. does it work? is it an improvement?). However, March and Smith (1995) argued that DSR should be concerned both with utility, as a design science, and with theory, as a natural science explaining how and why IT systems work within their operating environments. March and Smith (1995) found that DSR contribution lies in the novelty of the artifact and in the persuasiveness of the claims that it is effective. Along the same thread, Markus, Majchrzak, and Gasser (2002) found that a new IS design theory was required for a class of user requirements called emergent knowledge processes (EKPs), which are defined as patterns of organizational activity that exhibit three characteristics in combination: “deliberations” with no best structure or sequence; highly unpredictable potential users and work contexts; and information requirements that include general, specific, and tacit knowledge distributed across experts and non-experts.

From a different view on design theory, Hevner, March, Park, and Ram (2004) indicated that DSR is informed by both existing theory (produced by natural or behavioral science research) and by identified business needs. According to Hevner et al. (2004), such theories explain or predict organizational and human phenomena related to the identified business need and inform researchers and practitioners of the interactions among people, technology, and organizations that must be managed if an information system is to achieve its stated purpose, namely improving the effectiveness and efficiency of an organization. Hevner et al. (2004) further noted that DSR is a problem solving paradigm and knowledge and understanding of a problem domain and its solution are achieved in the building and evaluation of an IT artifact to meet the identified business need. The goal of behavioral science research is truth. The goal of design science research is utility. Hevner et al. (2004) argued that truth and utility are inseparable. Truth informs design and utility informs theory. According to Gregor & Jones (2007), Hevner et al. (2004) argue with the use of the word “theory” for design type knowledge, preferring to restrict the word to the possibly more familiar natural science (and, later, social science) types of theory.

According to Venable (2006), design theory building is a central activity related to problem diagnosis, technology invention or design (to solve problems), and technology evaluation. Venable (2006) indicated that theory building occurs before, during, throughout, and at the end as a result of Design Science Research. Venable argues that design theory should be in the form of utility theories, which relate improvements expected from applying a particular type or types of technologies to a particular type of problem. During the same year, Gregor (2006) examined the structural nature of theory in the discipline of Information Systems and proposed a taxonomy for classifying developed theories. Using the primary goals of theory (analysis, explanation, prediction, and prescription), Gregor (2006) distinguished five interrelated types of theory: (1) theory for analyzing; (2) theory for explaining; (3) theory for predicting; (4) theory for explaining and predicting (EP); and (5) theory for design and action. The theory for design and action says how to do something. It is about the principles of form and function, methods, and justificatory theoretical knowledge that are used in the development of IS. Models and methods can be evaluated for completeness, simplicity, consistency, ease of use, and the quality of results obtained through use of the method.

According to Gregor and Jones (2007), IS design theory allows the prescription of guidelines for further artifacts of the same type and that design theories can be about artifacts that are either products or methods. As the word “design” is both a noun and a verb, a theory can be about both the principles underlying the form of the design and also about the act of implementing the design in the real world. According to Gregor & Jones

(2007), researchers in design science have tended not to speak of theory in relation to design knowledge at all, but have focused more on design research as an activity that results in artifact construction.  One year later, based on Hevner et al.'s (2004) work, Peffers, Tuunanen, Rothenberger, and Chatterjee's (2008) developed a design science research methodology (DSRM) resulting from theory that incorporates principles, practice rules, and procedures required to carry out such design science (DS) research and a mental model for its presentation.  DSRM may support with the recognition and legitimization of DS research and its objectives, processes, and outputs and it should help researchers to present research with reference to a commonly understood framework, rather than justifying the research paradigm on an ad hoc basis with each new paper.

According to Gregor and Hevner (2013), theory is only one form that a DSR contribution can take.  They argued that contributions to knowledge could be partial theory, incomplete theory, or even some particularly interesting and perhaps surprising empirical generalization in the form of a new design artifact.  Based on Gregor and Hevner's (2013) findings, what is likely to be the most critical part of a DSR article is how the author stakes the claim to a knowledge contribution and provides convincing evidence that the research makes a practical contribution to the application context.

## Strengths and Weaknesses

Several studies exist in the literature review that are sound and support the problem statement and research questions. However, there are some studies that are less valuable and this section will endeavor to encapsulate both the strengths and weaknesses of some of the key studies related to the four main topics mentioned in the literature review: (a) wireless security; (b) need for security in free open public Wi-Fi hotspots; (c) client-side evil twin attack detection solutions; and (d) DSR principles and methodology.

Supporting the topic of wireless security in general, Kirankumar et al. (2012) and Singh et al. (2014) indicate that wireless networks are more vulnerable to security attacks than wired networks, due to the broadcast nature of the transmission, and that despite the productivity, convenience and cost advantage that WLAN offers, the radio waves used in wireless networks create a risk where the network can be hacked. NIST (2008) indicated that a passive security attack against WLAN such as "eavesdropping" allows the attacker to monitor wireless data transmissions between devices for message content, such as authentication credentials or passwords. An example of this attack is an attacker listening to transmissions on a WLAN between an AP and a client. Detecting evil twin access points is the first step in dealing with this problem.

Supporting the need for wireless security in free open public Wi-Fi hotspots, several authors such as Song et al. (2010, 2012), Hossen & Wenyuan (2014), Hsu et al. (2015) and Nakhila et al. (2015) emphasize the need for mechanisms to protect users against evil twin attacks that can severely compromise their security by making them more vulnerable to fraud and identity theft. Song et al. (2010, 2012) described an evil twin attack in a wireless LAN as a hard- or software-based 802.11 rogue Wi-Fi access point (AP) that looks like a legitimate one offered on the premises, but actually has been set up by a hacker to "eavesdrop" all wireless communications done by the victims. Han et al. (2009, 2011), Nikbakhsh et al. (2012), and Kim et al. (2012) indicate that the growing popularity of WLANs, has increased the risk of evil twin attacks and the lack of knowledge and awareness of this threat possessed by users make this issue extremely disturbing. Monica

& Ribeiro (2011) indicate the importance of detecting evil twin attacks to prevent attacker’s effective interception of all kinds of sensitive data such as passwords or credit card information.

The client-side evil twin access point detection studies indicated strength based on systems that have been recently developed by academic researchers. Hossen and Wenyuan (2014) and Nakhila et al. (2015) developed client-side solutions, CETAD and SSL/TCP protocol-based, for the most popular evil twin attack scenario where the attacker’s evil twin AP uses broadband cellular service, e.g. 3G/4G LTE, to access the Internet. Based on the review of the literature, Hossen & Wenyuan and Nakhila et al. are the only existing studies that assume the attacker using a different gateway from a legitimate AP. Evil twin attacks that use a different gateway from a legitimate AP (mobile attacks) will become more popular nowadays due to the increase in the Internet access speed of mobile connections, such as 3G/4G LTE, and the inclusion of mobile hotspot capabilities in virtually all new mobile devices (Szongott et al., 2012; Nakhila et al., 2015). Unfortunately, there is limited research focused on client-side solutions that will allow wireless users to verify the authenticity of access points at free open public Wi-Fi hotspots and protect themselves from mobile evil twin attacks. CETAD was the only solution evaluated at a large scale in public Wi-Fi hotspots.

Early studies conducted by Han et al. (2009, 2011) and Song et al. (2010, 2012) are considered weak because they developed a client-side detection system based on timing measurements which are mainly characterized for technology dependency and low efficiency impacting detection results. Nikbakhsh et al. (2012) developed a client-side detection solution based on traceroute results that can be captured by an attacker and send to the wireless client using the rogue wireless network. Han et al. (2009, 2011) and Szongott et al. (2015) require previous knowledge of the public Wi-Fi network. Additionally, in studies by Han et al. (2009, 2011), Song et al. (2010, 2012), Monica and Ribeiro (2011), Nikbakhsh et al. (2012), Kim et al. (2012), and Hsu et al. (2015), RTT-based, ETSniffer, WiFiHop, traceroute, multiple signal detection systems, and ET Detector provided limited client-side detection targeted only to the specific evil win attack scenario where the attacker uses the legitimate AP for Internet access instead of a more popular scenario where the attacker uses a different gateway from a legitimate AP such as broadband cellular service, e.g. 3G/4G LTE, to access the Internet. Studies by Song et al. (2010, 2012), Monica and Ribeiro (2011), and Szongott et al. (2015) had problems providing client-side evil twin detection solutions that require to install a server within the hotspot LAN, the implementation of a script in a service provider hosting service, or extra Wi-Fi devices. Han et al. (2009, 2011), Song et al. (2010, 2012), Monica & Ribeiro (2011), Nikbakhsh et al. (2012), Kim et al. (2012), Lanze et al. (2014), Hossen & Wenyuan (2014), Nakhila et al. (2015), Hsu et al. (2015), and Szongott et al. (2015) developed client-side detection systems that could not distinguish which AP is evil twin and which is legitimate, and as result could not offer the user to connect to a legitimate AP after detection. Additionally, all existing client-side approaches assume that the user has not connected to the target public Wi-Fi network in the past. Also, none of the studies protect the user for the duration of the Wi-Fi connection, discovering and reporting on new mobile evil twin access points. Protection is only provided to the user at the beginning based on the assumption that the attacker will be connected when the user connects to the public Wi-Fi network. Lastly, existing solutions were evaluated in a lab environment and in the field (universities, cafes, restaurants, and airports) using their own evil twin APs. Researchers did not evaluate their solution in hotel public Wi-Fi environments and did not aim at detecting real evil twin APs.

Several strengths and a weakness related to the problem and research question in this dissertation report are identified in the design science research section of the literature review. The strengths associated with the literature review of design theory are that Walls et al. (1992) and Markus et al. (2002) both used IS design theories targeted to develop executive information systems (EIS) and systems to support emergent knowledge process (EKPs), respectively. Hevner et al. (2004) defined the limitations of design science within the IS discipline via a conceptual framework for understanding IS research and established a set of guidelines for conducting, evaluating and presenting DSR. Gregor (2006) and Venable (2006) underscores the role and structural nature of theory in design science research. Peffers et al. (2008) addressed the lack of a methodology to serve as a framework for carrying out DS research in information systems and a template for its presentation.

The weakness related to the literature of design theory was identified in the study by March and Smith's (1995) contending that in order to insure IT research is both relevant and effective, both design science and natural science activities are needed. In addition, the study by Gregor and Hevner (2013) was largely concentrated on presenting practical guidance on how to comprehend, position, and present DSR knowledge contributions and publishing unrelated to this study's problem.

### Gaps in the Literature

Most of the studies presented in the literature review for this dissertation report did not include a thorough review of the existing client-side evil twin attack detection solutions and limitations. Therefore, the artifacts developed as part of these studies are not robust since they are based on a very limited number of requirements and assumptions. In addition, there are no client-side evil twin attack detection studies that are based on DSR principles and methodology. The design science research literature helped with the creation of DSR principles, procedures and specifications supported the artifact construction, implementation, and evaluation of the client-side evil twin attack detection system that is central to the study.

### Research Methods in Similar Studies

Peffers et al.'s (2008) DSRM has been adopted for this dissertation report. The DSR methodology is based on Hevner et al.'s DSR principles and includes the following process elements: (1) problem identification and motivation; (2) define the objectives for a solution; (3) design and development; (4) demonstration; (5) evaluation; and (6) communication. This study used the Peffers et al.'s (2008) research methodology as a model to extrapolate on various DSR approaches and presents on Table 1 a comparison of the process elements from methods in similar studies to Peffers et al.'s DSRM process elements. This comparison approach guided the creation of design specifications and procedures to develop, implement, and evaluate the client-side evil twin attack detection system at the center of this study.

Table 1

*Comparison of DSRM approaches*

| Peffers, Tuunanen, Rothenberger, and Chatterjee (2008) – design process elements | Walls, Widmeyer, and El Sawy (1992) | March and Smith (1995) | Markus, Majchrzak, and Gasser (2002) | Hevner, March, Park, and Ram (2004) | Gregor and Jones (2007) | Venable (2006) | Gregor and Hevner (2013) |
|---|---|---|---|---|---|---|---|
| 1. Problem identification and motivation | Kernel theories | Theorize | Characteristics of emergent knowledge processes (EKPs) | Important and relevant problems | Kernel theories<br>Purpose and Scope | Problem space | Purpose and Scope |
| 2. Objectives of a solution | | | Requirements for IT support of EKPs | Implicit in relevance | | Solution technology | Literature survey |
| 3. Design and Development | Design Method | Build artifact: constructs, model, method, and instantiation | EKP support system design and development principles for EKPs | Rigorous artifact iterative search process | Principles of implementation | Technology invention | Design artifact description and design search (development) process |

(continued)

| Peffers, Tuunanen, Rothenberger, and Chatterjee (2008) – design process elements | Walls, Widmeyer, and El Sawy (1992) | March and Smith (1995) | Markus, Majchrzak, and Gasser (2002) | Hevner, March, Park, and Ram (2004) | Gregor and Jones (2007) | Venable (2006) | Gregor and Hevner (2013) |
|---|---|---|---|---|---|---|---|
| 4. Demonstration | | | Effective EKP support system | Rigorous evaluation methods | Expository instantiation | | Novel artifact proof of concept demonstration |
| 5. Evaluation | Testable design process hypotheses | Evaluate artifact | | Evaluate | Testable propositions | Technology evaluation | Summative (final) testing |
| 6. Communication | | | | Communicate | | | Communicate |

## Synthesis of the Literature

Since the artifact design in this dissertation is based on wireless security, the need for security in free open public Wi-Fi hotspots, client-side evil twin attack detection solutions, and design science research principles and methodology involved in the artifact construction; the literature was required to be synthesized precisely related to the problem domain to provide a high-level, rational point of view.

Wireless security in general is strongly supported in studies conducted by NIST (2005), NIST (2008), Habibi et al. (2009), Feng (2012), Kirankumar et al. (2012), Kelly (2014), Szongott et al. (2012), and Singh et al. (2014). Their work describes existing solutions such as Wi-Fi Protected Access II (WPA2), Virtual Private Networks (VPN), and Universal Authentication Mechanism (UAM). However, according to Song et al. (2010, 2012), Monica & Ribeiro (2011), Choi et al. (2011), Cheng et al. (2013), Lanze et al. (2014), and Hossen & Wenyuan (2014), these solutions are not appropriate for protecting against evil twin attacks.

### *WPA2*

In personal mode, a pre-shared key (PSK) is established to encrypt traffic between client and AP. Such a mechanism can only protect against the evil twin attack if the PSK is hidden from the attacker. The PSK has to be supplied to potential users by some method, e.g., printed on a receipt. Therefore, in the case of public hotspots, the attacker can obtain the key by the same means as a public user and mount the attack unimpeded. In a public Wi-Fi environment, pre-shared keys are arduous to distribute and this differs with the hotspot' business goals (Choi et al., 2011; Lanze et al., 2014; Hossen & Wenyuan, 2014). In enterprise mode, the wireless AP acts as authenticator between a client and an authentication server using RADIUS and EAP. With EAP, a certificate authority (CA) certificate is to be used by devices to authenticate with the server before submitting credentials. Theoretically, evil twin attacks become impossible by this setup since the attacker cannot easily imitate the authentication server, as it is protected by strong cryptographic means. Nevertheless, there is a major weakness of this solution in practice. The mechanism has to be configured and carefully maintained by the operator, and operators of public hotspots in particular have no incentive to provide such a service. Furthermore, the validation of the server certificate by the client, the crucial element of the authentication process, is optional. If this is not done carefully by the user (i.e., the certificate check is activated and the user rejects the connection on seeing a certificate warning), imitation of the authentication server is possible, e.g., by harvesting and cracking handshakes. In addition, 802.1x needs a trustable authentication server to validate the wireless devices, which may not be feasible or suitable for the huge amount of traveling users to detect evil twin attacks by themselves in public Wi-Fi hotspots (Song et al., 2010, 2012; Choi et al., 2011; Lanze et al., 2014).

### *VPN*

VPNs become the standard when there is the requirement for connecting to the Internet through potentially untrustworthy wireless operators. Besides certificate-based attacks such as those on SSL, an attacker can terminate a VPN session (e.g., by dropping management packets) such that the connection returns to plain mode, usually without a noticeable notification to the user. The use of VPN solutions, is much more complex in terms of implementation and still leaves users susceptible to layer 2 and denial-of-service attacks. A user can configure their wireless device to setup a VPN connection through a public access point and all the traffic between the wireless device and the AP will be encrypted. However, VPN technology is not easily accessible for all users since such security service providers usually charge a monthly fee (Monica & Ribeiro, 2011; Cheng et al., 2013; Lanze et al., 2014).

### *UAM*

Free public Wi-Fi hotspots commonly provide a UAM. Usually, the initial URL accessed by the user is redirected to a captive portal, a website hosted by the operator that provides a disclaimer requiring the acceptance of the terms of use. However, the attacker can easily emulate this sort of page. UAM at hotspots does not allow the user to confirm the integrity of the hotspot or its provider. Hence, this method does not offer any security at all for the user pertaining to the evil twin attack (Lanze et al., 2014).

The need for security in free open public Wi-Fi hotspots is strongly supported in studies conducted by Han et al. (2009, 2011), Kim et al. (2012) and Nikbakhsh et al. (2012). Their findings indicate that as people's expectation of free open public Wi-Fi availability increases, the security of such networks becomes more important increasing the risk of wireless security attacks. According to JiWire's Mobile Audience Insights Report Q4 2013, nearly 85% of U.S. public Wi-Fi hotspots are free. Since the goal of free open public Wi-Fi hotspots is to provide convenience and to attract customers, security mechanisms are not in place.

Hossen and Wenyuan's (2014) study findings indicate that public Wi-Fi provides free, open, and zero liability Internet access to customers. However, generally consumers are oblivious to the danger on public Wi-Fi networks, such as evil twin attacks, causing identity theft, hacking, and breeched bank accounts (Private Wi-Fi, 2013). Public Wi-Fi users need to protect themselves from such threats. Particularly, a study conducted by The Guardian in 2011, launched two evil twin attacks conducted with volunteers, in which they successfully gather users' usernames, passwords, messages and even credit card information. This study reinforced that many public Wi-Fi hotspots have no forms of identification, except their wireless network names (SSID), which can be easily impersonated.

According to Harris poll (2014), 39% of U.S. adults have accessed or transmitted sensitive information while on public Wi-Fi without taking any steps to protect their data. Table 2 presents ways in which adults have accessed sensitive information while using public Wi-Fi.

Table 2

*Ways in which adults have accessed sensitive information while using public Wi-Fi*

| | Activity | Percentage |
|---|---|---|
| 1. | Say they have checked a bank account | 26% |
| 2. | Say they paid a bill | 19% |
| 3. | Say they have sent an email with sensitive information such as their Social Security number or an account number | 8% |
| 4. | Say they have filed their taxes | 6% |
| 5. | Say they have done so in another way | 10% |

The survey conducted by Harris poll (2014) also revealed U.S. adults' perceptions and attitudes toward potential threats when accessing free public Wi-Fi. This survey proves that despite their concern over the potential threats that public Wi-Fi poses, many users still perform activities that could make them vulnerable to identity theft. Table 3 shows U.S. adults' perceptions of potential threats when accessing free public Wi-Fi.

Table 3

*U.S. adults' perceptions of potential threats when accessing free public Wi-Fi*

| | Potential threat | Percentage |
|---|---|---|
| 1. | U.S. adults mentioned identify theft | 88% |
| 2. | Answered compromised accounts | 76% |
| 3. | Noted that fraudulent tax filings could be a potential issue | 39% |

Based on the literature and surveys, there is evidently a need for security on free open public Wi-Fi networks. Most of the public Wi-Fi hotspots are open, free and do not have security protections in place against wireless security attacks (Monica & Ribeiro, 2011; Hossen & Wenyuan, 2014; Nakhila et al., 2015). Wi-Fi's popularity makes it an attractive target for attackers to compromise and to eavesdrop wireless client information since many Wi-Fi hotspot users are unaware of the hidden risks that the technology poses, such as evil twin attacks, making users vulnerable to fraud and identity theft. Users enjoy the benefits of free open public Wi-Fi; however, they are not able to differentiate the ones that are safe from the ones that are not. Wi-Fi users must assume the responsibility for device protections in the light of these types of attacks.

Several researchers have been exploring detection methods of evil twin attacks for free open public Wi-Fi networks. However, existing solutions are mainly for network administrators instead of wireless users. According to Hossen & Wenyuan (2014), administrator-side solutions are not applicable to public Wi-Fi hotspots and more feasible in environments such as infrastructure networks, e.g. corporate networks. Kim et al.'s (2012) study indicated that recently several evil twin AP detection methods have been designed in order to overcome the administrator-side problems in a client-side solution. However, Kim et al. stated that existing client-side solutions have a cumbersome process in detecting fake APs in practice. Similarly, Lanze et al. (2014) indicated that existing solutions have limitations regarding requirements, ease of deployment, attacker model, and detection efficacy.

Han et al. (2009, 2011), Song et al. (2010, 2012), and Hossen & Wenyuan (2014) discovered that a client-side evil twin detection method based on timing measurements (e.g. RTT, IAT) is able to distinguish a one-hop from a multihop setting. However, Monica & Ribeiro (2011), discovered that timing measurement methods are technology dependent. According to their study, with the increase in wireless networks speeds, transmission delay differences between a wireless node and a wired node will eventually fade. This means that a multihop setting may become indistinguishable from a one-hop setting. Nakhila et al. (2015) indicated that timing-based methods need to monitor many packets in order to obtain accurate measurement, which makes the evil twin attack detection take a longer time to complete. In addition, Nakhila et al. (2015) stated that timing-based detection will be unreliable when the attacker uses a faster Internet connection such as broadband cellular service as the evil twin AP. The detection system developed as part of this study is not based on timing.

Han et al. (2009, 2011), Song et al. (2010, 2012), Monica & Ribeiro (2011), Nikbakhsh et al. (2012), Kim et al. (2012), Lanze et al. (2014), Hossen & Wenyuan (2014), and Hsu et al. (2015) assume in their studies that the attacker will use the legitimate wireless network gateway to pass through client data traffic (multihop attack). Nakhila et al. (2015) found that their detection methods will fail especially when the attacker uses a faster Internet connection (i.e. cellular broadband connection) compared to the legitimate wireless network. The attacker can delay the response time of the transmitting packets between the server and the wireless client to match the transmission time of the packets passing through the legitimate AP (Nakhila et al., 2015). Szongott et al. (2012), Hossen & Wenyuan (2014), and Nakhila et al. (2015) further indicated that evil twin attacks that use their cellular broadband connection will become more popular nowadays due to the increase in the Internet access speed of mobile connections, such as 3G/4G LTE or WiMAX and the inclusion of mobile hotspots capabilities in virtually all new mobile devices. The detection system developed as part of this study protects users from attackers that utilize a different gateway from a legitimate access point (mobile attack).

Han et al.'s (2009, 2011) and Song et al.'s (2010, 2012) client-side evil twin detection methods rely on existing networking protocols to work and can be executed by end users without any help from network administrators. Similarly, Monica & Ribeiro (2011), Kim et al. (2012), Nikbakhsh et al. (2012), Lanze et al. (2014), Hsu et al. (2015), Hossen & Wenyuan (2014), Szongott et al. (2015), and Nakhila et al. (2015) developed secure client-side evil twin detection methods that do not require network administrator privileges or network administrator assistance from hotspots networks. According to their studies, client-side evil twin detection methods must not require any administrative access to modify the routers or wireless access points. There is no need to modify the network architecture, hardware or software on either client or server side applications. Furthermore, Monica & Ribeiro (2011), as well as Kim et al. (2012), Hossen & Wenyuan (2014), Nakhila et al. (2015), and Szongott et al. (2015) indicated that the client-side evil twin detection system must be an automated application for whenever the user joins a public Wi-Fi hotspot. The detection system developed as part of this study does not require network administrator assistance or privileges and is automated with no intervention from users to ensure usability.

As mentioned previously, Han et al. (2009, 2011) developed a detection method that calculates the round trip time between the wireless user and the DNS server to independently determine whether an AP is legitimate or not without wireless administrator assistance. However, Song et al. (2010, 2012) found that since this work mainly utilizes the training detection technique and uses a relatively static threshold to differentiate normal and malicious scenarios, it needs to pre-gather the information of the target wireless network. Song et al. (2010, 2012) further indicated that Han et al.'s (2009, 2011) method could not be applied to those traveling users at the client side, since once the traveling users are in different areas, the network situation may have significantly changed. The trained knowledge in one wireless network can be hardly applicable to another network. Additionally, Szongott et al.'s (2015) system will not work if the user connects to a public Wi-Fi network for the first time since it requires previous knowledge of the network to assist the user. Recent studies conducted by Hossen & Wenyuan (2014) and Nakhila et al. (2015) also found that to detect an evil twin AP, the system should neither require any training knowledge of the target wireless network nor depend on the types of wireless networks to guarantee free open public hotspots.  The detection system developed as part of this study does not require knowledge of the wireless hotspots infrastructure, AP list and/or user/hosts (trained knowledge) and works on any type of IEEE 802.11-based wireless networks.

From an equipment requirement perspective, Song et al.'s (2010, 2012) study presented an evil twin detection solution that require to install a server within the hotspot LAN with ETSniffer for measuring Inter-packet arrival time (IAT) and detecting an evil twin AP.  In addition, this custom server must be available for the solution to work properly.  Monica & Ribeiro's (2011) method requires the implementation of a script in a service provider hosting service.  Hsu et al.'s (2015) method requires an additional Wi-Fi device with no sensitive data to assist with the detection. Szongott et al. (2015) requires an external server to store learned data.  A recent study conducted by Hossen & Wenyuan (2014) indicated that a client-side detection solution must be able to verify an access point in a hotspot and thus cannot assume any custom infrastructure support (e.g. hardware or software).  Hossen & Wenyuan (2014) further stated that designing an infrastructure-side solution would require hotspot providers to re-design existing hotspots, which is unlikely to happen because most hotspots are free services with no independent revenue.  According to Han et al.' (2009, 2011), Monica & Ribeiro' (2011), Kim et al.' (2012), Nikbakhsh et al.' (2012), Lanze et al.' (2014), Hossen & Wenyuan' (2014), and Nakhila et al.' (2015) studies, to guarantee usability and availability to the client, a client-side detection method must discover evil twin APs using their Wi-Fi enabled devices (e.g. laptops, smartphones, tablets, etc.) without additional equipment.

Hossen & Wenyuan’ (2014) and Nakhila et al.’ (2015) methods do not have the rigid requirement of having a custom server inside the LAN, rather their study leverage a public web server. However, their method does not cover the scenarios when the attacker blocks access to the public website. Furthermore, Hossen & Wenyuan’s method did not cover the scenario when the attacker presents an invalid certificate while ISP information is retrieved from the public website. The detection system developed as part of this study does not require any additional equipment or custom infrastructure support, leverages public servers, addresses blocked public website and invalid certificate scenarios, and works on any type of Wi-Fi enabled devices.

According to Han et al. (2009, 2011), Song et al. (2010, 2012), Monica & Ribeiro (2011), Kim et al. (2012), Nikbakhsh et al. (2012), Lanze et al. (2014), Hsu et al. (2015), Szongott et al. (2015), Hossen & Wenyuan (2014), and Nakhila et al. (2015), a client-side evil twin detection solution must warn the end user of an evil twin attack immediately in real time, before any data is transmitted, to avert being exposed to the attacker in the least, even when the attack may last for a short period of time. Song et al. (2010, 2012) found that evil twin attacks are hard to trace, because they can suddenly and randomly be launched and shut down, and last only for a short time after the attacker achieves his goal. Nakhila et al. (2015) indicated that once the attack has been detected, it is very challenging to identify which AP is rogue and which is legitimate because both provide Internet access that could have comparable quality. Furthermore, Hossen & Wenyuan (2014) claims that after the attack has been detected, their method allows the wireless users to connect to a legitimate AP. However, Hossen & Wenyuan (2014) did not include this in their algorithm. Additionally, Hossen & Wenyuan (2014) and Nakhila et al. (2015) assume that all the APs will be detected in the initial wireless network scanning and that they will be able to associate to all the APs in the public Wi-Fi network, which in practice is not always the case. All the studies assume that the attacker is already in the hotspot and is connected when the wireless user connects to the public Wi-Fi network. None of the studies addresses the case where the attacker appears later in the hotspot. Existing solutions do not protect the public Wi-Fi users for the duration of the Wi-Fi connection, discovering and reporting on new mobile evil twin access points. Additionally, all existing client-side approaches assume that the user has not connected to the target public Wi-Fi network in the past. According to Kumar & Paul (2016), the operating system stores the SSID and BSSID with which it was previously connected to in the client's preferred network list, and it is always in the exploration of the same and whenever detects attempts to connect to it. Therefore, the client will automatically connect to a potential evil twin AP when using the public Wi-Fi hotspot. The detection system developed as part of this study warns the wireless user of an evil twin attack before any traffic is transmitted and in addition, after evil twin detection, the system connects the user to a legitimate AP. The detection system also protects the users while they are connected to the public Wi-Fi network. The detection system protects the users when they have connected to a previous target network in the past. Lastly, the detection system protects the user in the case that not all the APs are detected in the initial wireless network scanning and when the client is not able to associate to all the APs in the public Wi-Fi network.

Han et al.' (2009, 2011), Song et al.' (2010, 2012), Monica & Ribeiro' (2011), Kim et al.' (2012), Nikbakhsh et al.' (2012), Lanze et al.' (2014), Hsu et al.' (2015), Szongott et al.’ (2015), Hossen & Wenyuan’ (2014), and Nakhila et al.’ (2015) solutions work with free open public Wi-Fi networks. Monica & Ribeiro (2011) and Hsu et al. (2015) indicated in their studies that the evil twin attack is usually launched at public places where open Wi-Fi networks are available. Public Wi-Fi hotspots are ideal networks as there is no way for the users to distinguish rogue from legitimate APs (Abdollah, 2007). The detection system developed as part of this study works with free open (unencrypted) public Wi-Fi networks.

Han et al. (2009, 2011) and Song et al. (2010, 2012) indicated that client-side evil twin detection methods must be resistant to environment change and consider influencing factors such as network saturation and receive signal strengths fluctuation. If the workload of the legitimate AP is extremely heavy, this may adversely affect the response time and lead to incorrect rogue AP detection. The time difference between legitimate and evil twin scenarios becomes less distinguishable. According to Song et al. (2010, 2012), when multiple devices synchronously attempt to send packets to the same AP, medium access collisions emerge and spur the phenomenon of network saturation. This phenomenon stochastically increases the time for transmitting packets from a client to the AP. According to Monica & Ribeiro (2011), in multi-hop wireless networks, especially with high traffic load, packet losses are frequent. The detection system developed as part of this study is technology independent (e.g. received signal strength fluctuation, network saturation, or network traffic conditions).

Hossen & Wenyuan (2014) and Nakhila et al. (2015) assumed that the BSSID of a hotspot legitimate AP is unique and use that as a reference to switch between different APs with the same SSID in the hotspot. According to Szongott et al. (2015), Lanze et al.

(2014), and Kumar and Paul (2016), SSIDs and BSSIDs can easily be spoofed by an attacker as the legitimate AP always transmits the SSID and the BSSID.  Further, Nakhila et al. (2015) method will not work when the evil twin AP is in a different subnet from the legitimate AP.  The detection system developed as part of this study considers the scenario when the attacker uses the same SSID, BSSID, and subnet of a hotspot legitimate AP.

Most of the existing client-side evil twin attack detection solutions have been evaluated at small scales and in lab environments.  Additionally, Han et al. (2009, 2011) evaluated their method at two universities in the United States and China.  Song et al. (2010, 2012) evaluated their method at one university in the United States.  Monica & Ribeiro's (2011) method was evaluated at one university.  Kim et al.'s (2012) study was evaluated at cafes and universities but details were not provided.  Nikbakhsh et al.'s (2012) method was evaluated in neither a lab nor the field.  Lanze et al.'s (2014) method was evaluated solely in a lab.  Hsu et al. 's (2015) method was evaluated at a university.  Hossen & Wenyuan (2014) evaluated their method at thirty locations, among them restaurants, cafes, universities and airports in the United States and China.  Nakhila et al.'s (2015) method was evaluated in a lab.  Szongott et al.'s (2015) evaluated their method in a lab using a simulator with real-world data. All the studies used their own evil twin APs on evaluation.  None of the studies have been evaluated in hotel public Wi-Fi environments and aimed at detecting real evil twin APs.  The detection system developed as part of this study was evaluated extensively in a lab and at a hotel (public Wi-Fi hotspot) in Ecuador.  The detection system aimed at detecting real evil twin APs.  Since no mobile evil twin APs were detected in the wild during the field evaluation period, the author proceeded to evaluate the artifact with the mobile evil twin AP used in the lab.

Han et al.' (2009, 2011), Song et al.' (2010, 2012), Monica & Ribeiro' (2011), Kim et al.' (2012), Hossen & Wenyuan' (2014), and Hsu et al.' (2015) solutions were evaluated for performance (effectiveness and efficiency). Hossen & Wenyuan's (2014) ISP-based solution was based on the attacker using cellular broadband connection as the network gateway. Hossen & Wenyuan (2014) did not measure the artifact's effectiveness in a controlled environment, but instead in an uncontrolled environment. In their study, Hossen & Wenyuan (2014) provided comprehensive metrics for the performance evaluation of their client-side detection method using the following standard metrics: (a) *Accuracy* indicates how accurately the method detects evil twin AP attacks; (b) *Precision* is the fraction of positively detected attacks to all positively detected attacks (correctly of incorrectly); and (c) *Recall* (also called in the literature True Positive Rate or TPR) is the fraction of positively detected attacks to all attacks that should be positively detected. To calculate these metrics, Hossen & Wenyuan (2014) used True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The TP and TN represent correct classification, and FP and FN represent incorrect classification. In regards to efficiency, Hossen and Wenyuan's time delay analysis included both detection of evil twin APs that use mobile Internet as the access network for connecting to the Internet and detection of evil twin APs that use the legitimate AP for Internet access. Additionally, only DHCP configuration time information was provided. Association time was not included in their calculations. Also, time information on the rest of the algorithm steps were not provided. Hossen & Wenyuan claimed that connection to a legitimate AP was included in their time delay analysis; however, Hossen & Wenyuan did not include this step in their detection algorithm. Information on all the factors impacting efficiency was not included in their time delay analysis. Lastly, it is not clear whether the time delay calculation included data collection, detection, and connection to a legitimate AP after detection; data collection and detection; or only detection. Nakhila et al.'s time delay technique provided a complete list of measurements and factors impacting efficiency. The detection system developed as part of this study was evaluated for performance effectiveness using Hossen & Wenyuan (2014)'s evaluation technique and Nakhila et al.'s technique was leveraged to improve upon Hossen and Wenyuan's to measure time delay.

Han et al. (2009, 2011), Song et al. (2010, 2012), Monica & Ribeiro (2011), Lanze et al. (2014), Nakhila et al. (2015), and Hsu et al. (2015) developed their client-side detection solutions using a laptop platform. Han et al. (2009, 2011), Monica & Ribeiro (2011), and Nakhila et al. (2015) used Linux OS. Hsu et al. (2015) used Windows 7. Nakhila et al. (2015) used C language. Szongott et al. (2015) used Java. Kim et al. (2012) and Hossen & Wenyuan (2014) developed their client-side detection solutions using a smartphone platform with Android OS. The client-side evil twin attack detection system discussed in this study was constructed in a prototype environment to support public Wi-Fi users. The detection system central to this study was built on a laptop platform with Linux OS and Java.

Appendix A shows existing studies requirements and limitations mapping based on the literature review.

Building an effective client-side evil twin attack detection artifact required that procedures and specifications based on DSR principles be developed to guide the successful construction, implementation, and evaluation of this type of artifact. These DSR principles that are grounded in design theory are important because according to Venable (2006), theory building occurs before, during, throughout, and at the end and as a result of DSR. According to Venable (2006), theory building in DSR begins with the spark of an idea, a nascent concept for a not-yet-existing (or not-yet-applied) technology as the solution for a problem or type of problem. This spark of an idea may come from (1) recombining ideas and conceptualizations of problem spaces; (2) realizing new possibilities for solutions; (3) recombining existing solutions/technologies; (4) imagining new technologies; and (5) realizing new applications for existing technologies.

Hevner et al.'s (2004) study states that artifacts are not exempt from behavioral theories. To the contrary, the creation of design artifacts relies on existing kernel theories that are applied, tested, modified, and extended through the experience, creativity, intuition, and problem solving capabilities of the researcher (Markus et al., 2002; Walls et al., 1992).

According to Gregor and Jones (2007), an IS design theory is something in an abstract world of man-made things, including abstract ideas such as models and algorithms. Gregor and Jones (2007) further indicate that a design theory instantiated would have a physical existence in the real world. According to their research, theories for design and action continue to be highly influential in IS, despite the fact that they are not always recognized as theories. Gregor and Jones (2007) stated that the main the characteristic of theories for design and action is that they focus on "how to do something" providing specific guidelines on how to design and develop an IT artifact such as a client-side evil twin attack detection artifact constructed as part of this study.  In their work, Gregor and Jones (2007) emphasized the importance of design work and design knowledge to be expressed as theory when building IT artifacts such as a client-side detection system.

The work of Gregor and Jones (2007) indicates that IS design theory shows the principles inherent in the design of an IT artifact that accomplishes some end, based on knowledge of both IT and human behavior. Gregor and Jones (2007) further indicate that as the word “design” is both a noun and a verb, a theory can be about both the principles underlying the form of the design and also about the act of implementing the design in the real world.  According to Gregor and Jones (2007), any design theory should include the following components: (1) the purpose and scope; (2) the constructs; (3) the principles of form and function; (4) the artifact mutability; (5) testable propositions; (6) justificatory knowledge; (7) principles of implementation; and (8) expository instantiation.  Table 4 describes each of the eight components of a design theory in the context of this study.

Table 4

*Eight Components of Information Systems Design Theory*

| Component | | Description |
|---|---|---|
| 1. | Purpose and Scope | The system will be used to provide traveling wireless users with a client-side detection tool to detect mobile evil twin attacks during their connection to free open public Wi-Fi networks. |
| 2. | Constructs | The system will help users detect mobile evil twin attacks and protect them during their connection to free open public Wi-Fi networks. |
| 3. | Principle of Form and Function | The system will be designed to detect and protect public Wi-Fi users from mobile evil twin attacks by providing them with a client-side detection tool. |
| 4. | Artifact Mutability | Suggestions for improving the system will be given for future work. |
| 5. | Testable propositions | How effective and efficient is the client side system in detecting mobile evil twin attacks in hotel public spaces? |
| 6. | Justificatory Knowledge | The proposed system will be based on design science theory from design sciences that provides an explanation for the design. |
| 7. | Principles of Implementation | The system will be implemented in a lab and in the field using the following steps: (1) establish system objectives; (2) define system functionality; (3) develop the system; and (4) evaluate the system. |
| 8. | Expository Instantiation | Examples of the client-side system in action will be provided to help explain the design and illustrate how the system function. |

Gregor and Hevner’s (2013) study indicates that theory is seen as an abstract entity, an intermeshed set of statements about relationships among constructs that aims to describe, explain, enhance understanding of, and, in some cases, predict the future (Gregor 2006). The type of theory that formalizes knowledge in DSR is termed design theory, the fifth of the five types of theory in Gregor’s taxonomy. This type of theory gives prescriptions for design and action, it says how to do something such as building a client-side evil twin detection artifact.

Peffers et al.’s (2008) design methodology would provide guidance for IS researchers to produce and present DS research in IS that is recognized as valuable, rigorous, and publishable in IS research outlets. For DS research, a methodology would include three elements: (a) conceptual principles to define what is meant by DS research; (b) practice rules; and (c) a process for carrying out and presenting the research. According to Gregor (2006), a design methodology can build on particular idiographic studies of what has worked in practice, on predictive relationships that are known but not fully understood, and on fully developed theories such as those relating to data representation or human behavior. Along the same thread, Gregor and Hevner (2013), stated in their study that the Peffers et al. research process offers a useful synthesized general model, building on other approaches.

Table 5 presents some examples of how DSR has been used in Security research.

Table 5

*How DSR has been used in IS*

| Article | Knowledge Contribution |
|---|---|
| Repairing trust in an e-commerce and security context: an agent-based modeling approach (Choi and Nazareth, 2014) | This study examines whether customers are willing to transact with an eCommerce vendor in light of security and trust violations. |
| A secure portable execution environment to support teleworking (James and Griffiths, 2013) | This study presents the design, development and trialing of the mobile execution environment (MEE), a secure portable execution environment designed to support secure teleworking. |
| Snakes and ladders for digital natives: information security education for the youth (Reid and Van Niekerk, 2013) | This study presents and evaluates a brain-compatible, information security educational game that can be used to introduce information security concepts to the youth from a very young age. |
| Secure activity resource coordination: empirical evidence of enhanced security awareness in designing secure business processes (D'Aubeterre, Singh, and Iyer, 2008) | This study examines the gap between systems development and systems security and develops an artifact that can be used to create business process models characterized by the secure exchange of information within and across organizational boundaries. |

**Summary**

In Chapter two, the study emphasizes the identification of literature that expounds on what is already known about the problem and synthesizing the literature to identify potential solutions that support the problem statement and research questions. The chapter began with the justification for the study by selecting papers for the review based on relevancy to design science, client-side evil twin detection systems, wireless security, and need for security in free open public Wi-Fi hotspots. The chapter then formed the summarization of existing studies based on the four main topic areas of (a) wireless security; (b) need for security in free open public Wi-Fi hotspots; (c) existing client-side evil twin detection solutions; and (d) DSR principles and methodology. The chapter further identified the strengths and weaknesses as well as gaps in the literature reviewed as they related to the four main topic areas and the problem statement. The overall goal of the literature review was met by synthesizing the foundational studies that were used to guide the development of a client-side evil twin attack detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi hotspots.  The design science research literature helped with the creation of DSR principles, procedures and specifications that supported the artifact construction, implementation, and evaluation of the client-side evil twin attack detection system that is central to the study.

# Chapter 3

# Methodology

## Overview of Research Methodology

To address the research problem and the methodology of how to accomplish the stated goal of designing and building a more effective, efficient, and practical client-side artifact to be used to detect mobile evil twin attacks, the author utilized a two phased research approach. In phase one, the author developed design principles, procedures and specifications to guide the design, construction, implementation, and evaluation of the prototype client-side evil twin detection artifact using Hevner's seven guidelines of DSR, Peffer's design science research methodology (DSRM), and Hossen & Wenyuan's (2014) study evaluation methodology. In phase two, the author extensively evaluated the performance of the client-side evil twin attack detection method by implementing a prototype system. The prototype system was implemented and evaluated in two environments. First, in a lab to analyze the requirements and demonstrate its effectiveness in a controlled environment. Second, in the field at a hotel public Wi-Fi hotspot to extensively evaluate the robustness of the system in practice. The prototype system aimed at detecting real mobile evil twin APs in the wild at a hotel property that provide free open public Wi-Fi in its public spaces. Since no mobile evil twin APs were detected during the field evaluation period, the author proceeded to evaluate the system with the mobile evil twin AP used in the lab. Hotel public Wi-Fi users spend a significant amount of time in hotel public spaces, also called social lobbies. Social lobbies define areas open to the public and contiguous to hotels' main lobbies. In these lobbies, hotels provide amenities and services like free Wi-Fi, comfortable chairs, waiter service, restaurant, a bar, and coffee shop (Kelley, 2012).

The techniques to evaluate the effectiveness and efficiency of the system were based on Hossen & Wenyuan's (2014) evaluation methodology which has been published and validated. Nakhila et al.'s technique was leveraged to improve upon Hossen and Wenyuan's to measure time delay. The client-side evil twin detection method developed as part of this dissertation was tested against Hossen & Wenyuan's (2014) method for detecting mobile evil twin attacks. The experiments aimed at showing that the detection system developed can detect mobile evil twin attacks more effectively and efficiently.

### *Design Science Research Guidelines*

To conduct, evaluate and present this research, the author used the seven guidelines for design science in information systems research developed by Hevner et al. (2004). The seven guidelines were reviewed and mapped to enable the development of the client-side evil twin attack detection system at the center of this study. The seven guidelines of Hevner et al. are based on the fundamental principle of design-science research that knowledge and understanding of a design problem and its solution are achieved in the building and application of a designed artifact in the form of a construct (vocabulary and symbols), model (abstractions and representations), method (algorithms and practices), or an instantiation (implemented and prototype systems). The seven design guidelines of Hevner et al. provide a structure to demonstrate the IS artifact via evaluation methods. Hevner et al.'s research indicates that the IT artifact defines the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, and evaluation of information systems can be effectively accomplished.

*Guideline 1: Design as an Artifact*

The result of a design-science IS research is a purposeful IT artifact created to address an important research problem. In support of guideline number one, this report developed a more effective, efficient, and practical client-side evil twin attack detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi hotspots. Additionally, it provided the framework to facilitate the design, implementation, and evaluation of an effective prototype client-side detection system to detect mobile evil twin attacks in hotel public Wi-Fi environments. A recent study conducted by Hossen & Wenyuan (2014) indicated that a client-side detection solution must be able to verify an AP in a hotspot and thus cannot assume any custom infrastructure support (e.g. hardware or software). Hossen & Wenyuan (2014) further stated that designing a solution with infrastructure support would require hotspot owners to modify hotspots, which is unlikely to happen because most hotspots are free services. According to Kim et al.' (2012) and Nikbakhsh et al.' (2012) studies, to guarantee usability and availability to the client, a client-side detection method must discover evil twin APs using their Wi-Fi enabled devices (e.g. laptops, smartphones, tablets, etc.) without extra devices. Although this study leveraged some of the key features of Hossen and Wenyuan's (2014) method, its main focus was on improving its limitations with a novel approach. The prototype developed as part of this study is multi-vendor and open source.

### *Guideline 2: Problem Relevance*

The key objective of IS research is to acquire knowledge and understanding that enable the development of technology-based solutions to important and unsolved business problems. Design science delivers on this objective through the construction of innovative artifacts intended to change the phenomena that occur. The technology-based solution that addresses the problem in this report is the primary motivation of the study and potentially impacts wireless security protection in hotel free open public Wi-Fi since the artifact is specifically designed to help wireless users to independently detect mobile evil twin attacks. Thus, the IT artifact constructed as part of this study helps solve a business problem by equipping traveling users with a client-side detection system to protect themselves from mobile evil twin attacks in hotel free open public Wi-Fi networks.

### *Guideline 3: Design Evaluation*

The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well executed evaluation methods. According to Hevner et al. (2004), because design is inherently an iterative and incremental activity, the evaluation phase provides essential feedback to the construction phase as to the quality of the design process and the design product under development. Hevner et al. (2004) identified five design evaluation methods to evaluate artifacts: (a) observational (case study or field study); (b) analytical (static analysis, architecture analysis, optimization, or dynamic analysis); (c) experimental (controlled experiment or simulation); (d) testing (functional (black box) testing or structural (white box) testing); and (e) descriptive (informed argument or scenarios). To evaluate the artifact in depth in a hotel environment, the author used

Hossen & Wenyuan’s (2014) study evaluation methodology that has been published and validated.

*Guideline 4: Research Contributions*

Effective design-science research must provide clear contributions in the areas of the design artifact, design construction knowledge, and/or design evaluation knowledge. Design-science research holds the potential for three types of research contributions based on the novelty, generality, and significance of the designed artifact. Hevner et al. (2004) indicates that in a given research project, one or more of these contributions must be found: (1) the design artifact (it must enable the solution of unsolved problems and extend the knowledge base or apply existing knowledge in new and innovative ways); (2) foundations (the creative development of novel, appropriately evaluated constructs, models, methods, or instantiations); and/or (3) methodologies (the creative development and use of evaluation methods and new evaluation metrics).

Hevner et al. (2004) stated in their research that artifacts must accurately represent the business and technology environments used in research and must be “implementable”, hence the importance of instantiating design science artifacts. In other words, the artifact must demonstrate a clear contribution to the business environment, solving an important, previously unsolved problem. In this study, the artifact adds value to the hotels because it potentially provides a client-side evil twin detection solution to help users of hotel free open public Wi-Fi to independently detect mobile evil twin attacks and connect them only to legitimate APs while using Wi-Fi at hotel public spaces.

### *Guideline 5: Research Rigor*

Rigor addresses the way in which research is conducted. DSR requires the application of rigorous methods in both the construction and evaluation of the designed artifact. According to Hevner et al. (2004), rigor is derived from the effective use of the knowledge base and success is predicated on the researcher's selection of appropriate techniques to construct an artifact and the selection of appropriate means to evaluate the artifact. Hevner et al. (2004) indicates the construction of effective metrics is an important part of DSR and that researchers must constantly assess the appropriateness of their metrics. In this study, the artifact construction used design procedures and specifications based on DSR to provide Wi-Fi users with a client-side detection system to independently detect mobile evil twin attacks while connected to hotel public Wi-Fi spaces. After design and construction, the artifact was evaluated extensively for performance effectiveness in a lab and a hotel using the following performance metrics (Hossen & Wenyuan, 2014): (a) *Accuracy* indicates how accurately the method 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* (also called in the literature True Positive Rate or TPR) is the fraction of positively detected attacks to all attacks that should be positively detected. To calculate these metrics, the author used True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The TP and TN represent correct classification, and FP and FN represent incorrect classification.

### *Guideline 6: Design a Search Process*

Design is essentially a search process to discover an effective solution to a problem. Problem solving can be viewed as utilizing available means to reach desired ends while satisfying laws existing in the environment. Means are the set of actions and resources available to construct a solution. Ends represent goals and constraints on the solution. Laws are uncontrollable forces in the environment (Hevner et al., 2004). Effective design requires knowledge of both the application domain (e.g., requirements and constraints) and the solution domain (e.g., technical and organizational). In this study, the author described the search process in terms of iteratively identifying limitations in existing client-side detection solutions and creatively developing a solution to address them. The author employed design principles, procedures and specifications based on DSR to facilitate construction, implementation and evaluation of a client-side evil twin attack detection system.

### *Guideline 7: Communication of Research*

DSR must be presented both to technology-oriented as well as management-oriented audiences. According to Hevner et al. (2004), technology-oriented audiences need sufficient detail to enable the described artifact to be constructed and used within an appropriate organizational context. This allows end users to test and enjoy the benefits offered by the artifact and it enables researchers to build a cumulative knowledge base for further extension and evaluation. Additionally, the audiences should also understand the methods in which the artifact was constructed and evaluated. This creates repeatability of the research project and builds the knowledge base for further research extensions by design-science researchers in IS. Management-oriented audiences need sufficient detail to determine if the organizational resources should be committed to constructing and using the artifact within their specific organizational context (Hevner et al., 2004). The client-side evil twin detection system design, construction, implementation, and evaluation process developed in this study is communicated in the form of a solution manual attached as an appendix in this dissertation report.

## Specific Research Methods

### *Design Science Research Methodology*

The DRSM used to tackle the research problem as well as the design procedures and specifications used to construct the artifact in this study was resultant of design science principles using an organized method for building client-side detection architecture addressing the problem. Considering a mixing and condensing of design science process elements synthesized from the literature review, a set of design procedures and specifications were developed based on DSR principles and methodologies that were used to enable the construction, implementation, and evaluation of the client-side detection artifact. The overall goal was to evolve the emergent DSR into design application that was lifted and used to direct the construction and evaluation of the artifact.

According to Peffers et al. (2008), a DS research methodology would include three elements: conceptual principles to define what is meant by design science research, practice rules, and a process or procedure for carrying out and presenting the research. Hevner et al. (2004) introduced principles that define what DS research is, and what goals it should pursue, as well as practice rules (guidelines) that provide guidance for conducting it. The missing part was a process or procedure (methodology) that provides a generally accepted framework for carrying out research (Peffers et al., 2008).

Peffers et al.’s (2008) DSR methodology (DSRM) based on Hevner et al.’s DSR principles was adopted for carrying out this study. Peffers et al.’s (2008) DSRM incorporates principles, practices, and procedures and meets three objectives: it is consistent with prior DSR literature, it provides a nominal process model for doing DS research, and it provides a mental model for presenting and evaluating DS research in IS. According to Peffers et al. (2008), a mental model for the conduct and presentation of DS research will help researchers to conduct it effectively. Peffers et al. (2008) stated that “a mental model is a "small-scale model" of reality that can be constructed from perception, imagination, or the comprehension of discourse” (p. 10). Table 6 illustrates a model for the construction process of the artifact reported in this study guided by Hevner et al.’s DSR principles literature that was used to answer the study research questions. Figure 6 illustrates the design topology for this study.

Based on Peffers et al.’s DSRM, a set of design procedures and specifications were developed to facilitate the client-side evil twin attack artifact construction, implementation, and evaluation phases. Table 7 shows an outline of the knowledge base principles that were followed during each phase.

Table 6

*Artifact Construction Methods and Technologies Associated with Hevner's Design Principles*

| Hevner's design principles | Methods | Technologies |
|---|---|---|
| 1. The research must produce an artifact created to address a problem.<br>5. Rigor must be applied in both the construction of the artifact and its evaluation. | • Apply design principles to the design and construction of the prototype system<br>• Procure the network devices, laptops, smartphone, and software<br>• Assemble and interconnect devices based on design topology<br>• Install and configure Linux, Java SE Development Kit (JDK), Netbeans IDE, and Wireshark on client laptop<br>• Install and configure Kali Linux (Aircrack-ng) and Hostapd on ETA laptop<br>• Configure Android Mobile Hotspot & Tethering on smartphone<br>• Apply class C logical addressing scheme to devices across the topology<br>• Test connectivity through the use of the ping and traceroute utilities<br>• Develop client-side detection system<br>• Test algorithm and Repeat | • Visio<br>• Cisco Router<br>• Cisco LAN Switch<br>• Cisco Wireless Controller<br>• Cisco Access Points<br>• Lenovo laptops<br>• Motorola Android smartphone<br>• Linux Centos OS<br>• Java SE Development Kit (JDK)<br>• Netbeans IDE<br>• Wireshark packet analyzer<br>• Kali Linux (Aircrack-ng)<br>• Hostapd<br>• Android Mobile Hotspot & Tethering<br>• Cat5e cables<br>• USB wireless adapter<br>• Ping and Traceroute |

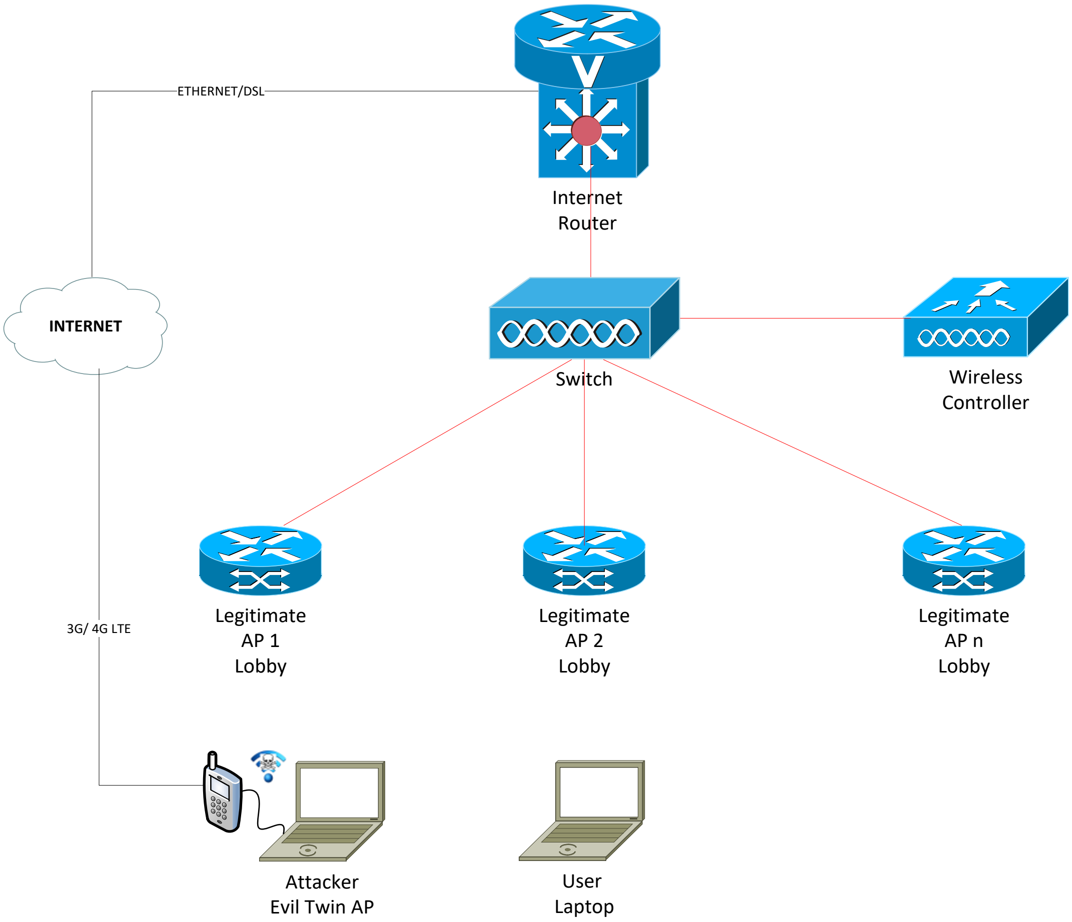


*Figure 6*. Design topology.

Table 7

*Peffers' DSRM Activity combined with Activity Description and Knowledge Base Principles*

| DSRM Activity | Activity Description | Knowledge Base Principles |
|---|---|---|
| 1. Problem identification and motivation | The need for client-side detection systems that will allow wireless users to protect against mobile evil twin attacks while using free open (unencrypted) public Wi-Fi. | Literature review to understand the problem's relevance, existing solutions, and limitations. |
| 2. Define the objectives for a solution | 1. The system must not be based on timing or traceroute.<br>2. The system must protect users from attackers that utilize a different gateway from a legitimate access point (mobile attack).<br>3. The system must not require network administrator assistance or privileges.<br>4. The system must be automated with no intervention from users.<br>5. The system must not require knowledge of wireless hotspots infrastructure, AP list and/or user/hosts (trained knowledge).<br>6. The system must not require any additional equipment or custom infrastructure support.<br>7. The system must leverage public servers. | Literature review to help define the objectives. |

(continued)

| DSRM Activity | Activity Description | Knowledge Base Principles |
|---|---|---|
| | 8. The system must work on any type of 802.11-based wireless networks.<br>9. The system must work with Wi-Fi enabled devices.<br>10. The system must work with free open (unencrypted) public Wi-Fi networks.<br>11. The system must be technology independent.<br>12. The system must protect the user when the attacker sets up the mobile evil twin AP with the same SSID, BSSID, and subnet of a legitimate AP.<br>13. The system must protect the user when the attacker blocks access to the public website used to get ISP information.<br>14. The system must protect the user when the attacker presents an invalid certificate while retrieving ISP information from public website.<br>15. The system must protect the user when not all the hotspots APs with the desired SSID are detected during the initial wireless network scanning and also when the client is not able to associate to all the APs in the public Wi-Fi network. | |

(continued)

| DSRM Activity | Activity Description | Knowledge Base Principles |
|---|---|---|
| | 16. The system must protect the users whether or not they have connected to a free open public Wi-Fi network in the past.<br>17. The system must protect the user while they are connected to the public Wi-Fi network.<br>18. The system must warn the user of an evil twin attack before any traffic is transmitted.<br>19. The system, after detection, must connect the user to a legitimate AP.<br>20. The system must be evaluated in the lab and in the field.<br>21. The system must aim at detecting real evil twin APs.<br>22. The system must be evaluated for performance using standard metrics. | |
| 3. Design and development | Design and construction of the system. | Application of principles, methods, and technologies to create the artifact. |
| 4. Demonstration | Demonstrate the use of the system in the lab. | Indicate how the system can be used in hotel environments to solve the problem. |

(continued)

| DSRM Activity | Activity Description | Knowledge Base Principles |
|---|---|---|
| 5. Evaluation | Evaluate the performance of the system at a hotel: How effective and efficient is the client-side system in detecting mobile evil twin attacks in hotel public spaces. | Evaluation technique from literature review to evaluate the artifact. |
| 6. Communication | Communicate the problem and its importance for replication in hotel environments. | Knowledge of hospitality environments related to client-side evil twin attack detection systems. |

The proposed design methodology presented on Figure 7 is based on Peffers et al.'s (2008) DSRM and was used as a model to document the design procedures and specifications that guided the construction of the client-side evil twin attack detection system in this study.

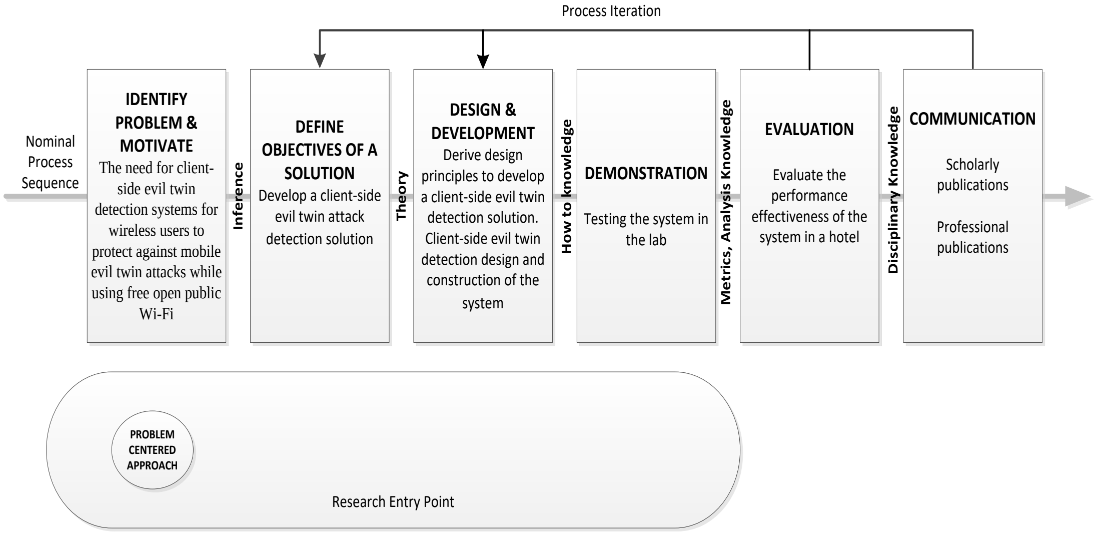

Process Iteration
Nominal Process Sequence
IDENTIFY PROBLEM & MOTIVATE
The need for client-side evil twin detection systems for wireless users to protect against mobile evil twin attacks while using free open public Wi-Fi
Inference
DEFINE OBJECTIVES OF A SOLUTION
Develop a client-side evil twin attack detection solution
Theory
DESIGN & DEVELOPMENT
Derive design principles to develop a client-side evil twin detection solution. Client-side evil twin detection design and construction of the system
How to knowledge
DEMONSTRATION
Testing the system in the lab
Metrics, Analysis Knowledge
EVALUATION
Evaluate the performance effectiveness of the system in a hotel
Disciplinary Knowledge
COMMUNICATION
Scholarly publications
Professional publications
PROBLEM CENTERED APPROACH
Research Entry Point

*Figure 7.* Proposed Design Methodology.

Based on Hevner et al.'s (2004) DSR principles 3 and 5, Table 8 presents the methods and techniques that were leveraged to collect and analyze data to show the effectiveness of the artifact.

Table 8

*Artifact Effectiveness Evaluation Methods Associated with Hevner's Design Principles*

| Hevner's design principles | Methods to collect data | Techniques to analyze data |
|---|---|---|
| 3. The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well executed evaluation methods.<br>5. Rigor must be applied in both the construction of the artifact and its evaluation. | • Evaluation Methodology (Hossen & Wenyuan, 2014)<br>• Wireshark packet analyzer<br>• Researcher-participant approach (Richey & Klein, 2007) | • Performance analysis metrics: (a) *Accuracy;* (b) *Precision*; and (c) *Recall* (Hossen & Wenyuan, 2014) |

## Design Procedures and Specifications

Based on previous chapters and sections of this study, design procedures and specifications derived from Hevner et al.'s (2004), Peffers et al.'s (2008), and Hossen & Wenyuan's (2014) including technologies, procedures, and techniques required to guide the design, construction, implementation, and evaluation of the client-side evil twin attack detection artifact otherwise known as CSMETAD (Client-Side Mobile Evil Twin Attack Detection) system at the center of this study are presented in Table 9.

Table 9

*Design Procedures and Specifications*

| Specifications | Procedures | Techniques | Technologies |
|---|---|---|---|
| 1. Design the artifact (physical and logical diagrams) | Apply DSR derived principles to guide the design of the client-side ETA detection system | Apply Industry Best Practice to design the client-side ETA detection system | Microsoft Visio |
| 2. Develop artifact specifications | Apply DSR derived principles to guide the development of the client-side ETA detection system specifications | Resources calculation for network, laptops and smartphone | Laptops, smartphone and network devices specifications |
| 3. Procure equipment for client-side evil twin attack detection system | Order equipment | Review equipment specs and pricing | MS Word (Bill of Materials) |

(continued)

| Specifications | Procedures | Techniques | Technologies |
|---|---|---|---|
| 4. Build and configure the client-side ETA detection system | Assemble and interconnect network devices<br>Install and configure Linux, JDK, Netbeans IDE and Wireshark on client laptop<br>Install and configure Kali Linux (Aircrack-ng) and Hostapd on ETA laptop<br>Configure Android Hotspot & Tethering on smartphone<br>Apply class C logical addressing scheme to devices across the topology<br>Develop client-side evil twin attack detection system | Apply Industry Best Practice to develop the client-side ETA detection system | Wireless client laptop<br>Linux, JDK, Netbeans IDE, Wireshark<br>ETA laptop<br>Kali Linux (Aircrack-ng), Hostapd<br>Smartphone<br>Android mobile hotspot & tethering<br>LAN Router<br>LAN Switch<br>Wireless Controller<br>Wireless Access Points<br>Ethernet cables<br>USB wireless adapter |
| 5. Test client-side ETA detection system | Test local connectivity and Algorithm | Network utility commands<br>Test cases | TCP/IP Utilities<br>Ping<br>Traceroute<br>Ifconfig<br>Netbeans IDE<br>System standard output |
| 6. Evaluate the performance effectiveness of the client-side ETA detection system in the lab and a hotel | Test algorithm and repeat | Researcher-participant approach<br>Monitor and analyze data captured using performance metrics | Netbeans IDE<br>System standard output<br>Wireshark packet analyzer |
| 7. Communicate the process for replication across Academia | Create and publish | Solution manual attached to dissertation appendix | Publish dissertation – Nova ProQuest |

## Artifact Design

The Client-Side Mobile Evil Twin Attack Detection (CSMETAD) system was designed based on Hevner's principle 5 through the application of rigorous design methods.

The design requirements, assumptions, and framework used to build the CSMETAD system were based on a thorough review of the literature and improved to address limitations in existing client-side evil twin attack detection solutions. The certified equipment included in Hossen & Wenyuan's (2014) study was replaced for the CSMETAD system to expand and provide protection to traveling users that utilize a different mobile platform and operating system in free open public Wi-Fi hotspots.

### *Design Requirements*

CSMETAD fulfills the following requirements:

1. It protects users from attackers that use a different gateway from a legitimate AP (mobile attack).
2. It protects users whether or not they have connected to a free open public Wi-Fi network in the past.
3. It protects users when not all the hotspot APs with the desired SSID are detected during the initial wireless network scanning.
4. It protects users when the client is not able to associate to all the APs in the public Wi-Fi network.
5. It protects users when the attacker sets up the mobile evil twin AP with the same SSID, BSSID (MAC address), and subnet of a legitimate AP.
6. It protects users when the attacker blocks access to the public website used to get ISP information.

7. It protects users when the attacker presents an invalid certificate while retrieving ISP information from a public website.
8. After mobile evil twin AP attack detection, it connects users to a legitimate AP.
9. It protects users for the duration of the public Wi-Fi connection, discovering and reporting on new mobile evil twin access points.
10. It is evaluated in the wild aiming to detect real mobile evil twin APs. In the case of not detecting real mobile evil twin APs during the field evaluation period, it is evaluated with the mobile evil twin AP used in the lab.
11. It is not based on timing or traceroute.
12. It does not require any additional equipment.
13. It does not require modification of the hotspot network infrastructure (custom infrastructure support).
14. It does not require trained knowledge of the target wireless hotspots infrastructure.
15. It is automated with no intervention from users.

*Design Assumptions*

The following are assumptions while designing CSMETAD:

1. The user may or may not have connected to the public Wi-Fi network in the past.
2. The wireless network client does not have any prior knowledge about the public Wi-Fi hotspot infrastructure.
3. The wireless network client may or may not able to detect all the public Wi-Fi hotspot APs with the desired SSID during the initial wireless network scanning.
4. The wireless network client may or may not be able to associate to all the APs in the public Wi-Fi network.

5. The public Wi-Fi hotspot provides free open (unencrypted) Wi-Fi access in its public spaces.
6. The public Wi-Fi hotspot supports a DHCP server that assigns dynamically network parameters to the clients (e.g. IP address, subnet mask, gateway, DNS, etc.).
7. The public Wi-Fi hotspot uses multiple AP architecture in which multiple APs support multiple wireless clients. All of the APs have the same SSID so that wireless users can automatically switch to another AP with a higher RSSI value when roaming across APs.
8. The public Wi-Fi hotspot does not use mobile Internet (e.g. 4G LTE).
9. The public Wi-Fi hotspot uses one ISP for Internet connectivity. The legitimate APs are connected to the same router sharing the same global IP address.
10. The public Wi-Fi hotspot has more than one AP installed in their public Wi-Fi space.
11. The public Wi-Fi hotpots APs have the same configuration (e.g. shared SSID, global IP address, DNS, etc.) to allow smooth AP association while the user roams throughout the public areas.
12. The public Wi-Fi hotspot requires acceptance of terms of use to be able to access the Internet.
13. ISP information of a legitimate AP and evil twin AP is different.
14. The attacker uses his laptop and smartphone with mobile AP functionality to launch an evil twin AP attack (mobile attack).
15. The attacker arrives later at the public Wi-Fi hotspot after the user has connected to a legitimate AP.

16. The attacker sets up the mobile evil twin AP with the same SSID, BSSID, and subnet of a legitimate AP.
17. The attacker disassociates the user from the legitimate AP and forces the user to connect to the mobile evil twin AP.
18. The attacker may block access to the public website used to get ISP information.
19. The attacker may present a valid or an invalid public website certificate while retrieving ISP information from public website.

### *CSMETAD Framework Overview*

The following provides an overview of the CSMETAD system. CSMETAD works in two phases:

- In (Phase 1) *data collection*, performs the initial wireless network scanning collecting data of all the access points (APs) in the public Wi-Fi hotspot. In this phase, CSMETAD categorizes the APs.
- In (Phase 2) *detection and protection*, detects mobile evil twin APs and connects the user to a legitimate AP. In this phase, CSMETAD protects the user for the duration of the public Wi-Fi connection discovering and reporting on new mobile evil twin APs.

*Phase 1 – data collection*

1. CSMETAD is designed based on the idea that the global IP addresses of two or more legitimate APs are the same, but they are different in the case of the legitimate APs and an evil twin AP. This occurs because the evil twin AP utilizes a different gateway than a legitimate AP (mobile attack) (Hossen & Wenyuan, 2014; Nakhila et al., 2015).
2. CSMETAD disables "auto-connections" to all public Wi-Fi networks protecting the user even when he or she has connected to a free open public Wi-Fi network in the past. The system iterates through all 802.11 wireless network connections, and after validating that the connection autoconnect is enabled and unencrypted, the system disables autoconnect. This requires for the wireless user to initialize CSMETAD before using the public Wi-Fi network.
3. CSMETAD scans the public Wi-Fi network to discover APs with selected SSID and adds APs with signal strength equal to or greater than -75 dBm to a list called "AP for selected SSID" list.
4. CSMETAD validates that the number of APs for the selected SSID is equal to or greater than 2. If CSMETAD determines that the number of APs for the selected SSID is less than 2, then CSMETAD displays message: "There is insufficient information to detect Evil Twin Attacks." CSMETAD ends. If CSMETAD determines that the number of APs for the selected SSID is equal to or greater than 2, then CSMETAD has sufficient information to detect ETAs.
5. CSMETAD goes through an AP iteration from "APs for selected SSID" list and associates to an AP, gets a Client DHCP address for the user, accepts terms of use to

access the Internet, and accesses secured public website to retrieve the global IP address of the AP. During the AP iteration process, if CSMETAD is not able to associate to an AP, then CSMETAD updates “APs for selected SSID” list with AP state as “unknown” and associates to the next AP on the “AP for selected SSID” list.

6. If CSMETAD is able to associate to an AP but not able to get a Client DHCP address for the user, accept terms of use to access the Internet, or access secured public website to retrieve the global IP address of the AP, then CSMETAD updates “APs for selected SSID” list with AP state as “unknown”, disassociates from current AP and associates to the next AP on the “APs for selected SSID” list.
7. If CSMETAD is able to associate to an AP, get a Client DHCP address for the user, accept terms of use to access the Internet, and access secured public website to retrieve the global IP address of the AP, then CSMETAD proceeds to verify that the public website certificate is valid.
8. CSMETAD is designed based on the idea that the attacker may present a valid or an invalid public website certificate while retrieving the global IP address from a public website. If CSMETAD determines that the public website certificate is invalid, CSMETAD updates “APs for selected SSID” list with AP state as “ETA”, adds AP MAC address to a list called “Learned ETA MAC address” list, disassociates from current AP and associates to the next AP on the “APs for selected SSID” list. If CSMETAD determines that the public website certificate is valid, only then, CSMETAD proceeds to determine the trusted global IP address to be used for the duration of the public Wi-Fi connection.

9. CSMETAD determines the trusted global IP address by selecting the global IP address with maximum number of occurrences. If CSMETAD determines that the global IP occurrences is less than 2, then CSMETAD displays message: “There is not enough information to categorize APs.”, updates “APs for selected SSID” list with AP state as “unknown”, and CSMETAD ends. If CSMETAD determines that the global IP occurrences is equal to or greater than 2, then CSMETAD proceeds to categorize APs.
10. CSMETAD categorizes APs by going through an AP iteration and validating whether the global IP address of an AP is the same as the trusted global IP address. If CSMETAD determines that the global IP address of an AP is not the same as the trusted global IP address, then CSMETAD categorizes the AP as “ETA” and add results to “Learned ETA MAC address” list. If CSMETAD determines that the global IP address of an AP is the same as the trusted global IP address, then CSMETAD categorizes the AP as “valid” and add results to the “APs for selected SSID” list.
11. CSMETAD moves to phase 2 after finishing AP iteration.

*Phase 2 – detection and protection*

1. CSMETAD re-scans the public Wi-Fi network to rediscover APs with selected SSID and adds APs with signal strength equal to or greater than -75 dBm to the “APs for selected SSID” list (new list).
2. CSMETAD validates that the number of APs for selected SSID is equal to or greater than 1. If CSMETAD determines that the number of APs for selected SSID is less than 1, then CSMETAD displays message: “Your device is out of range for the

selected public Wi-Fi hotspot. Please move closer". CSMETAD rescans the public Wi-Fi network.

3. If CSMETAD determines that the number of APs for selected SSID is equal to or greater than 1, then CSMETAD retrieves previously learned ETA MAC addresses from "Learned ETA MAC addresses" list and removes learned ETA MAC addresses from "AP for selected SSID" list.
4. After removal, CSMETAD validates that the number of APs for selected SSID is equal to or greater than 1. If CSMETAD determines that the number of APs for selected SSID is less than 1, then CSMETAD displays message: "You are located on the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi Hotspot". CSMETAD rescans the public Wi-Fi network. If CSMETAD determines that the number of APs for selected SSID is equal to or greater than 1, then CSMETAD starts iterating across all the APs in the "APs for selected SSID" list.
5. CSMETAD proceeds to validate APs even if they have the same SSID, MAC address, and subnet of a legitimate AP. CSMETAD goes through an AP iteration of "APs for selected SSID" list and associates to the AP with the highest signal strength. During the AP iteration process, if CSMETAD is not able to associate to an AP, then CSMETAD associates to the next AP on the "APs for selected SSID" list. If CSMETAD is not able to associate to any of the APs on the "APs for selected SSID" list, then CSMETAD rescans the public Wi-Fi network.
6. If CSMETAD is able to associate to the AP but not able to get a Client DHCP address for the user, confirm access to the Internet, or access secured public website to

retrieve the global IP address of an AP, then CSMETAD disassociates from current AP and associates to the next AP on the “APs for selected SSID” list.

7. If CSMETAD is able to associate to an AP, confirm Client DHCP address for the user, confirm access to the Internet, and access secure public website to retrieve the global IP address of an AP, only then, CSMETAD proceeds to verify that the public website certificate is valid.
8. If CSMETAD determines that the public website certificate is invalid, CSMETAD displays message: “CSMETAD has detected an ETA on the public Wi-Fi hotspot”, adds AP MAC address to a list called “Learned ETA MAC address” list (if ETA is not in the list), disassociates from current AP, and associates to the next AP on the “APs for selected SSID” list. If CSMETAD determines that the public website certificate is valid, only then, CSMETAD proceeds to get the global IP address of the AP.
9. If CSMETAD determines that the global IP address of an AP is not the same as the trusted global IP address, CSMETAD displays message: “CSMETAD has detected an ETA on the public Wi-Fi hotspot”, adds AP MAC address to “Learned ETA MAC address” list (if ETA is not in the list), disassociates from the current AP, and associates to the next AP with the highest signal strength on the “APs for selected SSID” list. If CSMETAD determines that the global IP address of an AP is the same as the trusted global IP address, then CSMETAD displays message: “Wi-Fi connection is safe. You are connected to a legitimate AP”. Then, CSMETAD waits for a disassociated wireless card event.

10. If CSMETAD receives a disassociated wireless card event, CSMETAD rescans the public Wi-Fi network to discover and report on new mobile evil twin access points for the duration of the public Wi-Fi connection. Algorithm phase 2 repeats (infinite loop).

The following is a list of replacement hardware and software included in the design of the CSMETAD system that is central to this dissertation report:

1. *Client Platform:* Hossen & Wenyuan's study's artifact was built for smartphone platforms. CSMETAD was built for laptop platforms. The client laptop platform for this study is a Lenovo Thinkpad laptop.
2. *Client Operating System (OS):* Hossen & Wenyuan's study's artifact was built for Android operating system. CSMETAD was built for Linux operating system. The Linux OS version for this study is 7.3.1611.
3. *Client Programming Language:* Hossen & Wenyuan's study's artifact's programming language was not provided in their study. CSMETAD was built using Java programming language. The Java SE Development Kit is version 1.8.0_131 (64 bits) and the NetBeans Integrated Development Environment is version 8.1.
4. *Mobile Evil Twin AP*: Hossen & Wenyuan (2014) performed the evaluation using a smartphone with mobile AP functionality as the evil twin AP (Nexus 4 Android smartphone with 3G data subscription and Android mobile hotspot and tethering). CSMETAD was evaluated using a laptop and smartphone with mobile AP functionality as the evil twin AP (Lenovo Thinkpad laptop, Kali Linux (Aircrack-ng) and Hostapd, Motorola Moto e$^5$smartphone with 4G data subscription and Android mobile hotspot and tethering).

## Artifact Specifications

### Wireless Client

The hardware and software specifications for the *wireless client* are described as follows:

*Hardware*

The Lenovo Thinkpad Laptop specifications can be retrieved from https://support.lenovo.com/mn/en/solutions/pd027202

*Software*

The Linux OS 7.3.1611 specifications designed for inclusion into the CSMETAD system can be retrieved from https://wiki.centos.org/Manuals/ReleaseNotes/CentOS7

The Java SE Development Kit 1.8.0_131 (64 bits) programming language specifications designed for inclusion into the CSMETAD system can be retrieved from https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf

The NetBeans Integrated Development Environment 8.1 specifications designed for inclusion into the CSMETAD system can be retrieved from https://netbeans.org/community/releases/81/relnotes.html

The Wireshark Packet Analyzer 2.0.0 software was installed in the client specifically for network packet analysis purposes.  The specifications can be retrieved from https://www.wireshark.org/docs/relnotes/wireshark-2.0.0.html

## Mobile Evil Twin AP

The hardware and software specifications for the *mobile evil twin AP* are described as follows:

### Hardware

The Motorola Moto e5 smartphone specifications can be retrieved from https://www.motorola.com/us/products/moto-e-plus-gen-5

The Lenovo Thinkpad Laptop specifications can be retrieved from https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-x/Thinkpad-X1-Carbon-4th-Gen/p/22TP2TXX14G

### Software

The Android Mobile Hotspot and Tethering specifications can be retrieved from https://www.verizonwireless.com/support/knowledge-base-217411/

The Kali Linux 4.14.0 (Aircrack-ng) specifications can be retrieved from https://www.kali.org/news/kali-linux-2018-1-release/

The Hostapd v2.7 specifications can be retrieved from http://w1.fi/hostapd/

## Lab Network

The hardware and software specifications for the *lab network* are described as follows:

The Cisco M10 router specifications documented in an installation guide (Cisco Systems, 2010) can be retrieved from http://downloads.linksys.com/downloads/userguide/1224655305646/Valet_Valet_Plus_M10_M20_UG_US_V10_D-WEB_3425-014530.pdf

The Cisco 3560 switch specifications documented in a spec sheet (Cisco Systems, 2009) can be retrieved from http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3560-series-switches/product_data_sheet09186a00801f3d7d.pdf

The Cisco 2504 wireless controller specifications documented in a spec sheet (Cisco Systems, 2016) can be retrieved from http://www.cisco.com/c/en/us/products/collateral/wireless/2500-series-wireless-controllers/data_sheet_c78-645111.pdf

The Cisco 3502I wireless access point specifications documented in a spec sheet (Cisco Systems, 2012) can be retrieved from http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1250-series/data_sheet_c78-594630.pdf

Figure 8 shows Logical Prototype Topology Design Diagram.

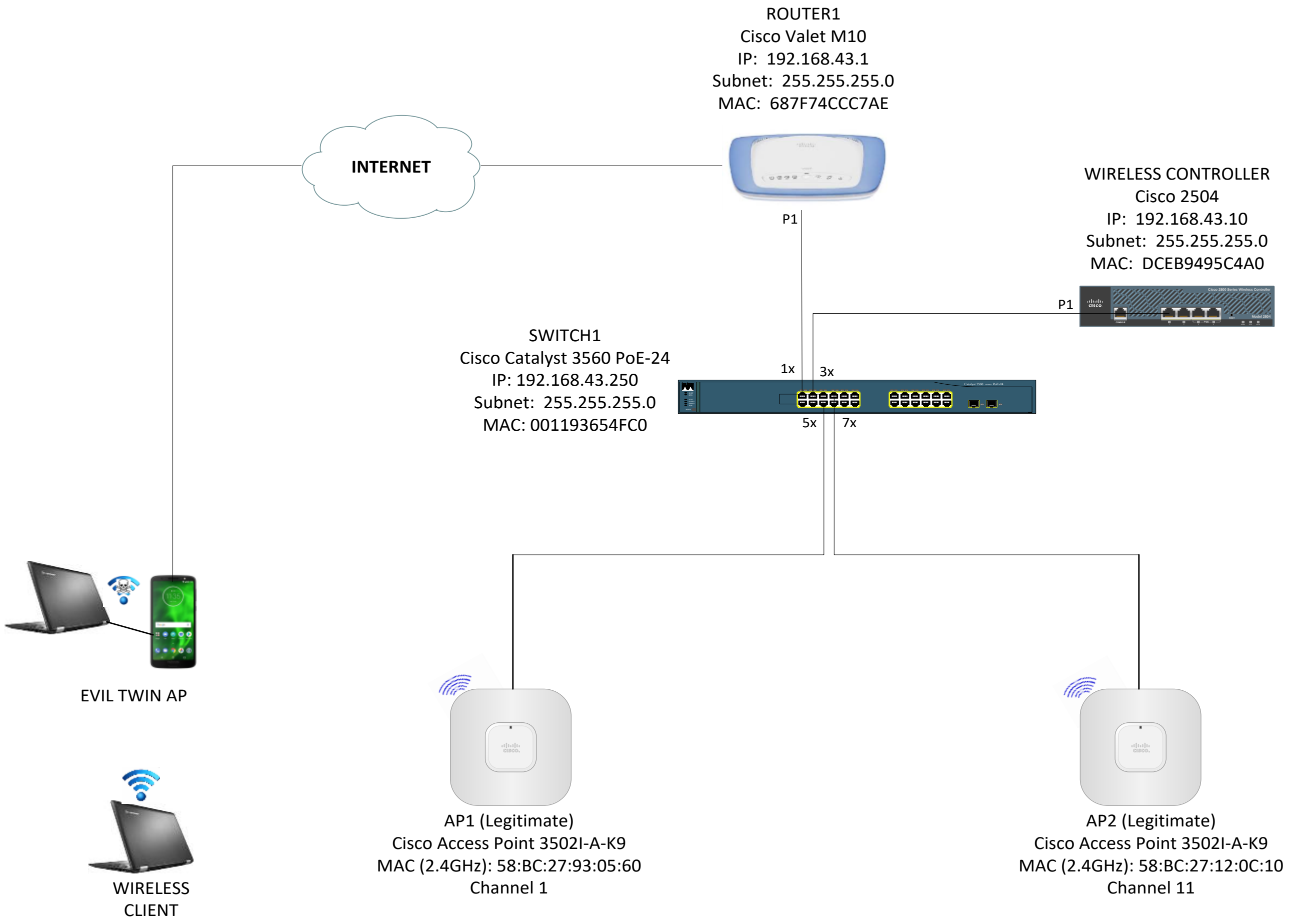


*Figure 8.* Logical prototype topology design diagram.

### Artifact Equipment Requirement

The components and costs to build the CSMETAD system and lab as defined in this dissertation report are listed in Table 10.

Table 10

*Evil Twin Detection Lab Environment Components and Costs*

| Component | Quantity | Estimated Cost |
|---|---|---|
| Hardware | | |
| Cisco 3560 Switch | 1 | $150 |
| Cisco Router M10 | 1 | $200 |
| Cisco Wireless Controller 2504 | 1 | $490 |
| Cisco Access Point 3502I-A-K9 (AP1) (legitimate AP) | 1 | $80 |
| Cisco Access Point 3502I-A-K9 (AP2) (legitimate AP) | 1 | $80 |
| Lenovo Laptop | 2 | $2,500 |
| Motorola Moto e$^5$ Android smartphone | 1 | $150 |
| USB wireless adapter | 1 | $40 |
| Ethernet cables | - | $30 |
| Software | | |
| Wireshark Packet Analyzer 2.0.0 | 1 | Free |
| Java SE Development Kit 1.8.0_131 | 1 | Free |
| NetBeans IDE 8.1 | 1 | Free |
| Linux Centos 7.3.1611 | 1 | Free |
| Kali Linux 4.14.0 (Aircrack-ng) | 1 | Free |
| Hostapd v2.7 | 1 | Free |
| Android Mobile Hotspot &Tethering | 1 | Free |
| Switch IOS | 1 | Included |
| Router IOS | 1 | Included |
| Controller IOS | 1 | Included |
| APs IOS | 3 | Included |
| Total | | $3,720 |

## Artifact Construction

Construction of the CSMETAD system was based primarily on Hevner's principles 1 and 5 through the creation of a viable artifact that relies on the application of rigorous construction methods.

*Lab Environment - Steps:*

1. Unpacking and assembling the equipment.
2. The Linux Centos 7.3.1611 OS was installed and configured in the Lenovo Thinkpad laptop (client) in accordance with the installation guide (Centos, 2016) retrieved from https://wiki.centos.org/Manuals/ReleaseNotes/CentOS7
3. The Java SE Development Kit 1.8.0_131 (64 bits) software was installed and configured in the Lenovo Thinkpad laptop (client) in accordance with the installation guide (Oracle, 2016) retrieved from http://docs.oracle.com/javase/8/docs/technotes/guides/install/index.html
4. The NetBeans Integrated Development Environment 8.1 software was installed and configured in the Lenovo Thinkpad laptop (client) in accordance with the installation guide (Netbeans, 2015) retrieved from https://netbeans.org/community/releases/81/install.html
5. The Wireshark Packet Analyzer 2.0.0 software was installed and configured in the Lenovo Thinkpad laptop (client) in accordance with the installation guide (Wireshark, 2014) retrieved from https://www.wireshark.org/docs/wsug_html/
6. The Kali Linux 4.14.0 (Aircrack-ng) was installed and configured in the Lenovo Thinkpad laptop (ETA) in accordance with the installation guide (Kali, 2018) retrieved from https://docs.kali.org/category/installation

7. The Hostapd v2.7 was installed and configured in the Lenovo Thinkpad laptop (ETA) in accordance with the installation guide (Hostapd, 2013) retrieved from https://w1.fi/hostapd/
8. The Motorola smartphone was configured with tethering in accordance with the instructions (Motorola, 2018) retrieved from https://www.verizonwireless.com/support/knowledge-base-217411/
9. The Cisco M10 router was installed and configured in accordance with the installation guide (Cisco Systems, 2010) retrieved from http://downloads.linksys.com/downloads/userguide/1224655305646/Valet_Valet_Plus_M10_M20_UG_US_V10_D-WEB_3425-014530.pdf
10. The Cisco 3560 switch was installed and configured in accordance with the installation guide (Cisco Systems, 2010) retrieved from http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3560/hardware/installation/guide/3560hig.pdf
11. The Cisco 2504 wireless controller was installed and configured in accordance with the installation guide (Cisco Systems, 2017) retrieved from https://www.cisco.com/c/en/us/support/docs/wireless/2500-series-wireless-controllers/113034-2500-deploy-guide-00.html
12. The Cisco 3502I access points was installed and configured in accordance with the installation guide (Cisco Systems, 2014) retrieved from http://www.cisco.com/c/en/us/td/docs/wireless/access_point/3500/quick/guide/ap3500getstart.pdf

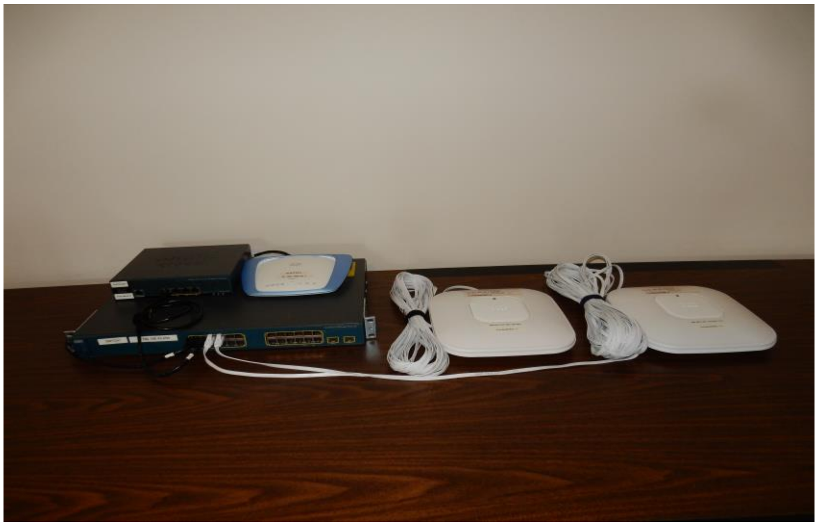


*Figure 9.* Lab network.

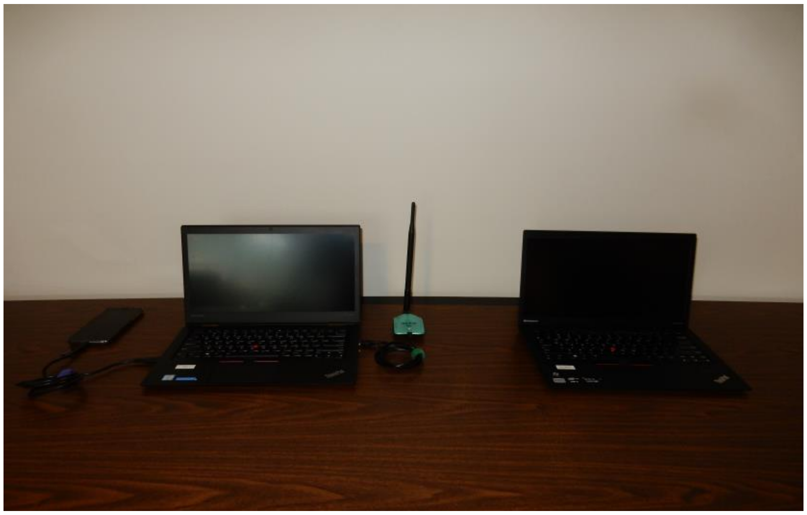


*Figure 10.* Lab wireless client and mobile evil twin AP.

*Client-side Mobile Evil Twin AP Detection (CSMETAD) Algorithm - Steps:*

*Phase 1: Data Collection*

**Basic Flow:**

1. System is initialized by the user before using the free open public Wi-Fi network.
2. System detects operating system.
3. System detects wireless network card.
4. System disables “auto-connections” to all public Wi-Fi networks.
5. System scans the public Wi-Fi network to discover available SSIDs (encrypted and unencrypted).
6. System creates list of **SSIDs in range**.
7. System presents SSIDs in range to the user.
8. User selects unencrypted public Wi-Fi hotspot SSID.
9. System creates list of **APs for selected SSID** with signal level equal to or greater than -75dBm.
10. System validates that the **number of APs for selected SSID** is equal to or greater than 2. **(Alternative Flow “a”)**
11. IF **number of APs for selected SSID** is equal to or greater than 2 THEN System displays message: “There is sufficient information to start detecting ETAs.”
12. System stops the network manager.
13. System activates wireless network card.
14. System starts iterating across all the APs in the **APs for selected SSID** list.
15. System associates to the AP in the **APs for selected SSID** list. **(Alternative Flow “b”)**

16. IF System is able to associate to the AP THEN System gets a Client DHCP address for the user. **(Alternative Flow “c”)**
17. IF System is able to get a Client DHCP address for the user THEN System accepts terms of use to access the Internet. **(Alternative Flow “d”)**
18. IF System is able to accept terms of use to access the Internet THEN System connects to secured public website to retrieve global IP address of the AP. **(Alternative Flow “e”)**
19. IF System is able to connect to the secured public website THEN System verifies that the public website certificate is valid. **(Alternative Flow “f”)**
20. IF the public website certificate is valid only THEN System is able to get the global IP address of the AP.
21. IF System ends iterating across APs in **APs for selected SSID** list THEN System validates that the **number of occurrences of a global IP address** is equal to or greater than 2. **(Alternative Flow “g”)**
22. IF the **number of occurrences of a global IP address** is equal to or greater than 2 THEN System has enough information to start categorizing APs and sets global IP as the trusted global IP address to be used for the duration of the public Wi-Fi connection.
23. System starts categorizing APs.
24. System starts iterating across all the APs in the **APs for selected SSID** list.
25. System validates that the global IP address for an AP is the same as the trusted global IP address. **(Alternative Flow “h”)**

26. IF the global IP address for an AP is the same as the trusted global IP address THEN System categorizes the AP as “valid”.  System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
27. IF System ends iterating across APs in **APs for selected SSID** list THEN System moves to Phase 2 detection and protection.

**<u>Alternative Flows:</u>**

*a) number of APs for selected SSID is less than 2*

On step 10 of the Basic Flow:

1. IF the **number of APs for selected SSID** is less than 2 THEN
2. System displays message: “There is insufficient information to detect ETAs.”
3. System ends.

*b) system is not able to associate to an AP*

On step 15 of the Basic Flow:

1. IF System is not able to associate to an AP THEN
2. System updates the **APs for selected SSID** list with the following results:
    a. Association status = false
    b. Client DHCP address = not detected
    c. Internet access = not detected
    d. Secured public website access = not detected
    e. Certificate status = not detected
    f. Global IP address = not detected
    g. AP state = unknown
3. System associates to the next AP in the **APs for selected SSID** list.

4. Flow of events returns to step 15 of the Basic Flow.
5. System ends.

*c) system is not able to get a Client DHCP address for the user*

On step 16 of the Basic Flow:

1. IF System is not able to get a Client DHCP address for the user THEN
2. System updates the **APs for selected SSID** list with the following results:
    a. Client DHCP address = not detected
    b. Internet access = not detected
    c. Secured public website access = not detected
    d. Certification status = not detected
    e. Global IP address = not detected
    f. AP state = unknown
3. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
4. Flow of events returns to step 15 of the Basic Flow.

*d) system is not able to accept terms of use to access the Internet*

On step 17 of the Basic Flow:

1. IF System is not able to accept terms of use to access the Internet THEN
2. System updates the **APs for selected SSID** list with the following results:
    a. Internet access = false
    b. Secured public website access = not detected
    c. Certification status = not detected
    d. Global IP address = not detected

    e. AP state = unknown

3. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
4. Flow of events returns to step 15 of the Basic Flow.

*e) system is not able to access secured public website*

On step 18 of the Basic Flow:

1. IF System is not able to access secured public website THEN
2. System updates the **APs for selected SSID** list with the following results:
    a. Secured public website access = false
    b. Certification status = not detected
    c. Global IP address = not detected
    d. AP state = unknown
3. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
4. Flow of events returns to step 15 of the Basic Flow.

*f) invalid certificate*

On step 19 of the Basic Flow:

1. IF System receives an invalid certificate message THEN
2. System updates the **APs for selected SSID** list with the following results:
    a. Certification status = invalid
    b. Global IP address = not detected
    c. AP state = ETA
3. System adds AP MAC address to the **learned ETA MAC address** list.

4. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
5. Flow of events returns to step 15 of the Basic Flow.

*g) number of occurrences of a global IP address is less than 2*

On step 21 of the Basic Flow:

1. IF System determines that number of occurrences of a global IP address is less than 2 THEN
2. System displays message: “There is not enough information to categorize APs”
3. System updates the **APs for selected SSID** list with the following result:
    a. AP state = unknown
4. System ends.

*h) global IP address for an AP is not the same as the trusted global IP address*

On step 25 of the Basic Flow:

1. IF System determines that the global IP address for an AP is not the same as the trusted global IP address THEN
2. System categorizes the AP as “ETA”.
3. System updates the **APs for selected SSID** list with the following result:
    a. AP state = ETA
4. System adds the AP MAC addresses to the **learned ETA MAC address** list.
5. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
6. Flow of events returns to step 25 of the Basic Flow.

**<u>Input and Output details:</u>**

1. **SSIDs in range** list = list of AP MAC address, signal level/RSSI, encryption status, frequency, and channel.  This list contains APs with encryption on and off.
2. **APs for selected SSID** list = list of AP MAC address, signal level/RSSI, encryption status, frequency, channel, Client DHCP address, Internet access, secured public website access, certification status, global IP address, and AP state.  This list contains only the APs with encryption off and signal/RSSI level equal to or greater than -75dBm.
3. **Learned ETA MAC address** list = list of ETA MAC addresses.

**<u>Rule details:</u>**

1. **Number of APs for selected SSID** = the number of APs for selected SSID must be equal to or greater than 2.
2. **Number of occurrences of a global IP address =** the number of occurrences of a global IP address must be equal to or greater than 2.

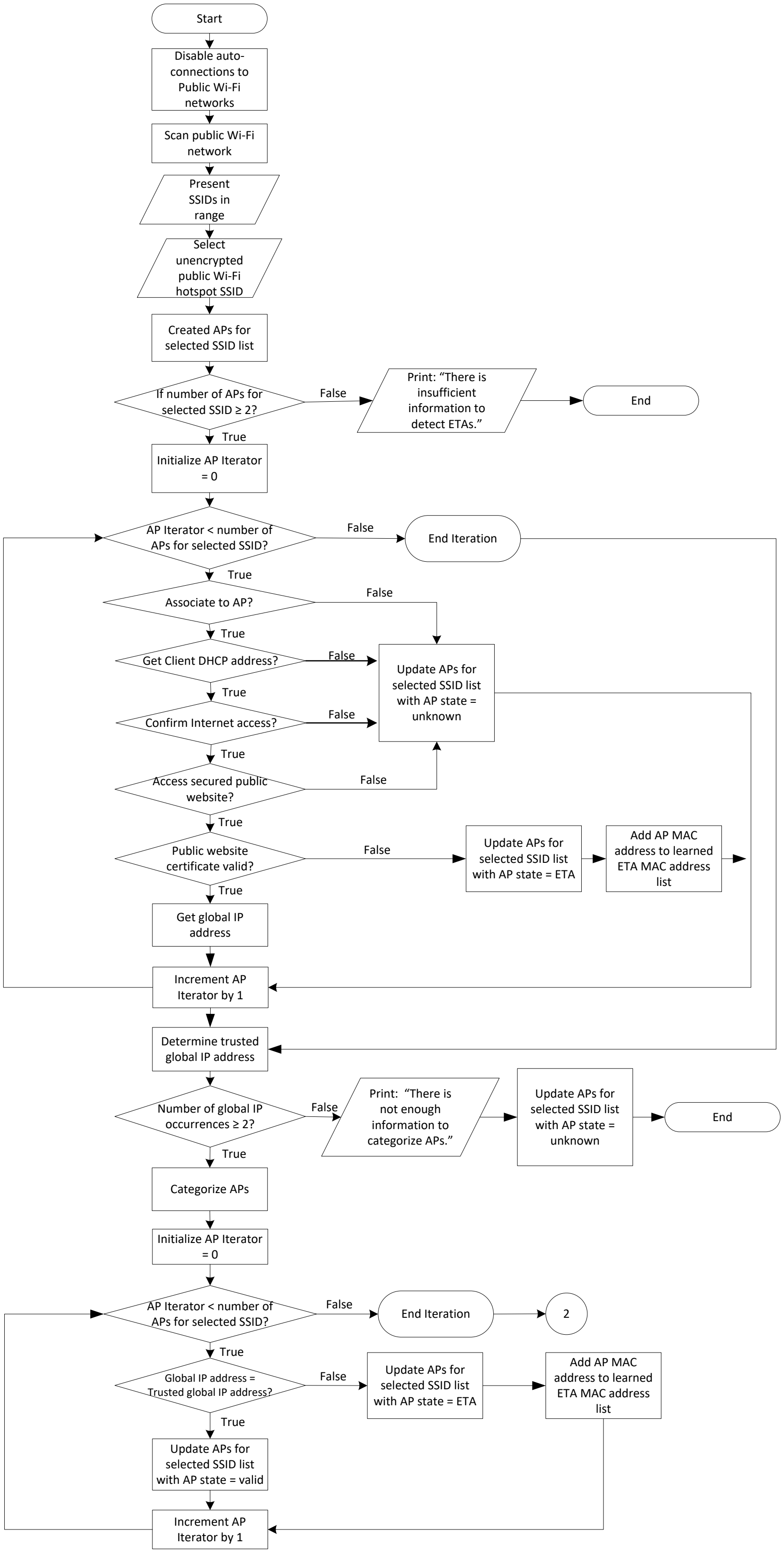

Start
Disable auto-connections to Public Wi-Fi networks
Scan public Wi-Fi network
Present SSIDs in range
Select unencrypted public Wi-Fi hotspot SSID
Created APs for selected SSID list
If number of APs for selected SSID ≥ 2?
False
Print: "There is insufficient information to detect ETAs."
End
True
Initialize AP Iterator = 0
AP Iterator < number of APs for selected SSID?
False
End Iteration
True
Associate to AP?
False
True
Get Client DHCP address?
False
Update APs for selected SSID list with AP state = unknown
True
Confirm Internet access?
False
True
Access secured public website?
False
True
Public website certificate valid?
False
Update APs for selected SSID list with AP state = ETA
Add AP MAC address to learned ETA MAC address list
True
Get global IP address
Increment AP Iterator by 1
Determine trusted global IP address
Number of global IP occurrences ≥ 2?
False
Print: "There is not enough information to categorize APs."
Update APs for selected SSID list with AP state = unknown
End
True
Categorize APs
Initialize AP Iterator = 0
AP Iterator < number of APs for selected SSID?
False
End Iteration
2
True
Global IP address = Trusted global IP address?
False
Update APs for selected SSID list with AP state = ETA
Add AP MAC address to learned ETA MAC address list
True
Update APs for selected SSID list with AP state = valid
Increment AP Iterator by 1

*Figure 11*. CSMETAD Algorithm Flow – Phase 1 Data Collection.

*Phase 2: Detection & Protection*

**Basic Flow:**

1. System rescans the public Wi-Fi network to rediscover APs with selected SSID.
2. System creates list of SSIDs in range.
3. System adds all the APs for selected SSID with signal level equal to or greater than -75 dBm to the **APs for selected SSID** list (new list).
4. System validates that number of APs for selected SSID is equal to or greater than 1. **(Alternative Flow "a")**
5. IF number of APs for selected SSID is equal to or greater than 1 THEN System retrieves learned ETA MAC addresses from **learned ETA MAC address** list. This list includes all ETA MAC addresses learned from the beginning of the program.
6. System removes learned ETA MAC addresses from **APs for selected SSID** list.
7. System validates that the **number of APs for selected SSID** is equal to or greater than 1. **(Alternative Flow "b")**
8. IF **number of APs for selected SSID** is equal to or greater than 1 THEN System starts iterating across all the APs in **APs for selected SSID** list.
9. Systems associates to the AP with the highest signal strength in the **APs for selected SSID** list. **(Alternative Flow "c")**
10. IF System is able to associate to the AP THEN System gets a Client DHCP address for the user. **(Alternative Flow "d")**
11. IF System is able to get a Client DHCP address for the user THEN System confirms access to the Internet. **(Alternative Flow "e")**

12. IF System is able to confirm access to the Internet THEN System connects to secured public website to retrieve global IP address of the AP. **(Alternative Flow "f")**
13. IF System is able to access secured public website THEN System verifies that the public website certificate is valid. **(Alternative Flow "g")**
14. IF the public website certificate is valid only THEN System is able to get the global IP address for the AP.
15. System validates that the global IP address for the AP is the same as the trusted global IP address. **(Alternative Flow "h")**
16. IF the global IP address for the AP is the same as the trusted global IP THEN the System displays message: "Wi-Fi connection is safe. You are connected to a legitimate AP".
17. System ends iterating across APs in **APs for selected SSID** list.
18. System waits for a disassociated wireless card event.
19. IF System receives a disassociated wireless card event THEN System proceeds to rescans the public Wi-Fi network. Algorithm phase 2 repeats (infinite loop).

**<u>Alternative Flows:</u>**

*a) number of APs for selected SSID is less than 1*

On step 4 of the Basic Flow:

1. IF the **number of APs for selected SSID** is less than 1 THEN
2. System displays message: "Your device is out of range for the selected public Wi-Fi hotspot. Please move closer".
3. Flow of events returns to step 1 of the Basic Flow.

*b) number of APs for selected SSID is less than 1*

On step 7 of the Basic Flow:

1. IF the number of APs with the selected SSID is less than 1 THEN
2. System displays message: “You are located on the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi Hotspot”.
3. Flow of events returns to step 1 of the Basic Flow.

*c) system is not able to associate to an AP*

On step 9 of the Basic Flow:

1. IF System is not able to associate to an AP THEN
2. System associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
3. Flow of events returns to step 9 of the Basic Flow.

*d) system is not able to get a Client DHCP address for the user*

On step 10 of the Basic Flow:

1. IF the System is not able to get a Client DHCP address for the user THEN
2. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
3. Flow of events returns to step 9 of the Basic Flow.

*e) system is not able to confirm access to the Internet*

On step 11 of the Basic Flow:

1. IF the System is not able to confirm access to the Internet THEN
2. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.

3. Flow of events returns to step 9 of the Basic Flow.

*f) system is not able to access secured public website*

On step 12 of the Basic Flow:

1. IF the System is not able to access secured public website THEN
2. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
3. Flow of events returns to step 9 of the Basic Flow.

*g) invalid certificate*

On step 13 of the Basic Flow:

1. IF the System receives an invalid certificate message THEN
2. System has detected an ETA on the public Wi-Fi network. System displays message: "CSMETAD has detected an ETA on the public Wi-Fi hotspot".
3. System adds the AP MAC address to the **learned ETA MAC address** list (if ETA is not in the list).
4. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
5. Flow of events returns to step 9 of the Basic Flow.

*h) global IP address for an AP is not the same as the trusted global IP address*

On step 15 of the Basic Flow:

1. IF the System determines that the global IP address for an AP is not the same as the trusted global IP address THEN
2. System has detected an ETA on the public Wi-Fi network. System displays message: "CSMETAD has detected an ETA on the public Wi-Fi hotspot".

3. System adds the AP MAC address to the **learned ETA MAC address** list (if ETA is not in the list).
4. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
5. Flow of events returns to step 9 of the Basic Flow.

**Input and Output details:**

1. **APs for selected SSID** list = list of AP MAC address, signal level/RSSI, encryption status, frequency, channel, Client DHCP address, Internet access, secured public website access, certification status, global IP address, and AP state. This list contains only the APs with encryption off and signal level equal to or greater than -75dBm.
2. **Learned ETA MAC address** list = list of ETA MAC addresses.

**Rule details:**

1. **Number of APs for selected SSID =** the number of APs for selected SSID must be equal to or greater than 1.

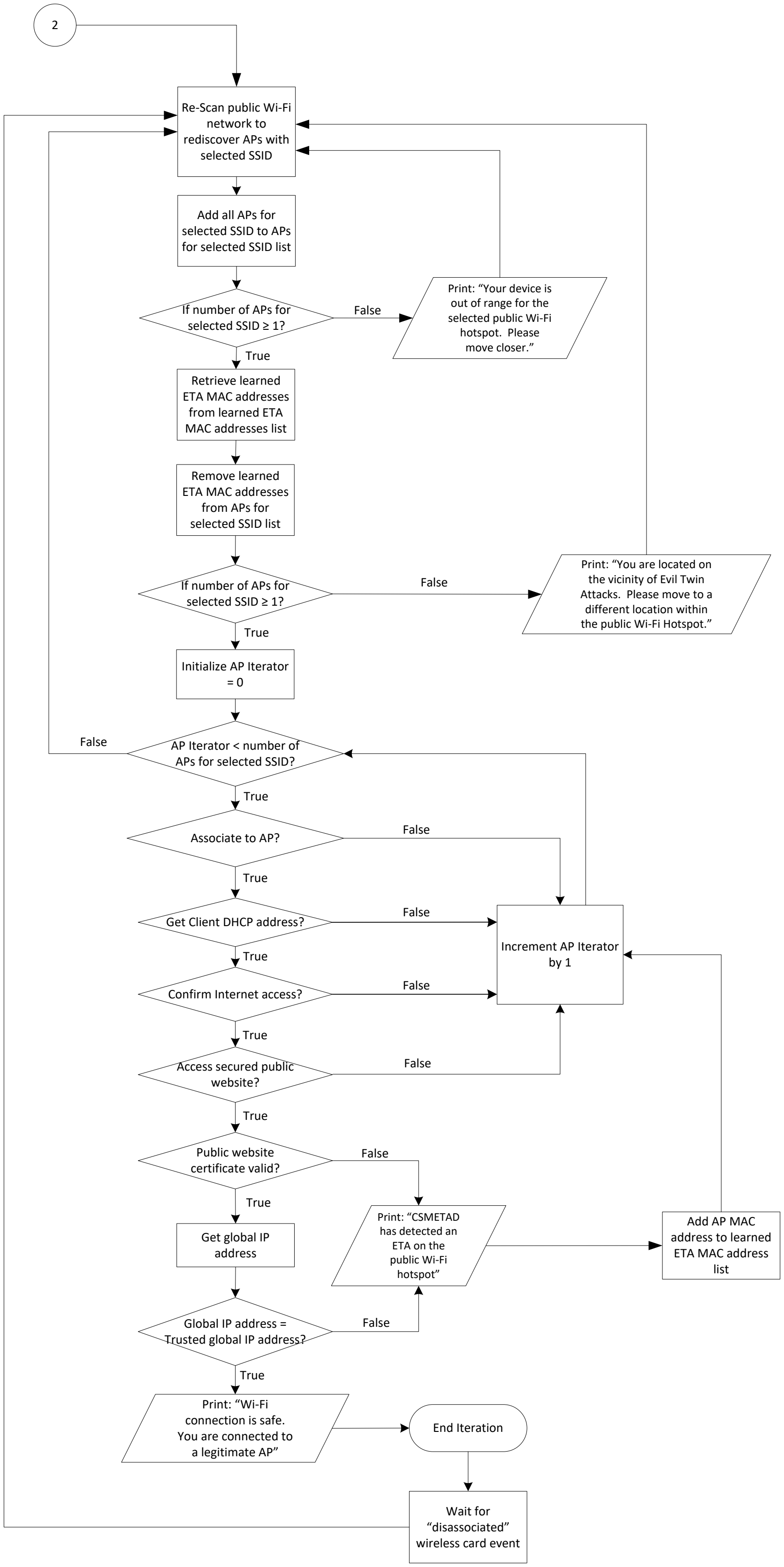

2
Re-Scan public Wi-Fi network to rediscover APs with selected SSID
Add all APs for selected SSID to APs for selected SSID list
If number of APs for selected SSID ≥ 1?
False
Print: "Your device is out of range for the selected public Wi-Fi hotspot. Please move closer."
True
Retrieve learned ETA MAC addresses from learned ETA MAC addresses list
Remove learned ETA MAC addresses from APs for selected SSID list
If number of APs for selected SSID ≥ 1?
False
Print: "You are located on the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi Hotspot."
True
Initialize AP Iterator = 0
False
AP Iterator < number of APs for selected SSID?
True
Associate to AP?
False
True
Get Client DHCP address?
False
Increment AP Iterator by 1
True
Confirm Internet access?
False
True
Access secured public website?
False
True
Public website certificate valid?
False
True
Print: "CSMETAD has detected an ETA on the public Wi-Fi hotspot"
Add AP MAC address to learned ETA MAC address list
Get global IP address
Global IP address = Trusted global IP address?
False
True
Print: "Wi-Fi connection is safe. You are connected to a legitimate AP"
End Iteration
Wait for "disassociated" wireless card event

*Figure 12*. CSMETAD Algorithm Flow – Phase 2 Detection & Protection.

## Artifact Testing

Rigorous testing took place with the CSMETAD system based on Hevner's DSR principle 5 in order to verify that the architecture components were working effectively according to the design.

### *Lab Environment Testing*

1. The TCP/IP utility "ifconfig" was used to verify the correct address configuration of the lab equipment, once the devices in the topology were connected, configured and developed in the construction phase.
2. The Cisco Operating System "show run" command was used to prove and troubleshoot the configuration of the router, switch, wireless controller, and wireless access points.
3. The TCP/IP utility "ping" was used to verify connectivity between router, switch, wireless controller, and wireless access points.
4. The TCP/IP utility "traceroute" was used to discover the path between devices across the topology.

Appendix B shows Algorithm test cases and results.

## Artifact Production

After the construction was complete, the CSMETAD system was brought into production mode. CSMETAD initially aimed at detecting real mobile evil twin AP attacks in the wild at a hotel property. Since no mobile evil twin APs were detected in the wild during the field evaluation period, the author proceeded to evaluate the system with the mobile evil twin AP used in the lab.

## Artifact Evaluation

The client-side mobile evil twin attack detection system was evaluated based on Hevner's principle 3 that asserts that the utility, quality, and efficacy of a design artifact must be rigorously validated via well executed evaluation methods. The author extensively evaluated the performance of the client-side evil twin attack detection method by implementing a prototype system. The prototype system was evaluated in two environments. First, in a lab to analyze the requirements and demonstrate the effectiveness in a controlled environment. Second, in the field at a hotel public Wi-Fi hotspot to extensively evaluate the robustness of the system in practice. The client-side mobile evil twin attack detection system aimed at detecting real mobile evil twin AP attacks in the wild at a hotel property that provide free open public Wi-Fi. Since no real mobile evil twin AP attacks were detected in the wild during the field evaluation period, the author proceeded to evaluate the system with the mobile evil twin AP used in the lab. Similar approach was used by Hossen & Wenyuan's study and the remainder of the client-side evil twin attack detection studies referenced in this dissertation.

The client-side evil twin detection method developed as part of this dissertation was tested against Hossen & Wenyuan's (2014) method for detecting mobile evil twin attacks. The experiments aimed at showing that the detection system developed can detect mobile evil twin attacks more effectively and efficiently.

The techniques to evaluate the effectiveness and efficiency of the system were based on Hossen & Wenyuan's (2014) evaluation methodology which has been published and validated and included the following:

1. Collected data from a hotel public Wi-Fi hotspot (public spaces).
2. Ran the experiments on both weekdays and weekends for a period of 5 weeks (2 weeks at the lab and 3 weeks at the hotel).
3. Collected approximately 300 hours of data.
4. Collected more than 151,000 instances of data.
5. Ran the detection system 140 times at the lab and 210 times at a hotel public Wi-Fi hotspot.
6. Monitored the network with Wireshark packet analyzer.

For efficiency, the author used Hossen and Wenyuan's technique to measure time delay but also leveraged Nakhila et al.'s technique to improve upon Hossen and Wenyuan's. Nakhila et al. included a complete list of measurements and factors impacting efficiency.

In this study, the author used a researcher-participant approach. According to Richey and Klein (2007), researchers are often the designer/ developers. In other words, by design they "go native" and observe themselves. "The researcher who ceases to be conscious of the observer role is said to be going native" (Singleton & Straits, 2005). In this study, the author participated as the user of the client-side evil twin attack detection system and the researcher observing the client-side evil twin attack detection system performance.

## Artifact Communication

The CSMETAD system design, implementation, and evaluation process including specifications and procedures is communicated as a solution manual named CSMETAD System Solution Manual and is included in Appendix D of the dissertation to be made available via ProQuest Dissertations database.

## Instrument Development and Validation

As a first phase in the assessment, and prior to continuing with testing the research questions in this study, the validity of the experiment was evaluated.  According to Albright and Malloy (2000), experimental validity is built on the way in which variables have an influence on both the outcomes of the research and the generality of research participants. Researchers have divided experimental validity into internal and external validity.

### *Internal Validity*

Internal Validity of a research study refers to the "extent to which its design and the data that it yields allow the researcher to draw accurate conclusions about cause-and-effect and other relationships within the data" (Leedy & Ormrod, 2005).  Similarly, according to Briggs & Schwabe (2011), internal validity is the question of whether the observed results were actually caused by the experimental treatment instead of by something else.

To establish internal validity, the researcher of this study examined Criterion Validity, also known as Instrumental Validity. According to Leedy & Ormrod (2005), instrumental validity is based on the premise that processes and instruments used in a study are valid if they parallel similar to those used in previous, validated research. Following Hossen & Wenyuan's (2014) validated research evaluation methodology, this study:

a) demonstrated the artifact in a laboratory setting;
b) evaluated the artifact in an uncontrolled environment;
c) built evaluation steps (such as when to run experiments and collect the data, what data to collect, how much data to collect, and how many times to run the tests at the hotspot);
d) analyzed performance (detection effectiveness) using standard metrics (accuracy, precision, and recall); and
e) used researcher-participant approach.

### *External Validity*

External validity of a research study refers to the "extend to which its results apply to situations beyond the study itself…the extent to which the conclusions drawn can be generalized to other contexts" (Leedy & Ormrod, 2005). Also, according to Briggs & Schwabe (2011), external validity is the degree to which results of the experiment would generalize to contexts other than those of the experimental conditions. Additionally, external validity is important to demonstrate that research results are applicable in natural settings, as contrasted with laboratory settings (King & He, 2005).

The researcher reached out to a hotel organization in Ecuador that provide free open public Wi-Fi in their public spaces and requested approval to participate in this study. Based on the responses, the hotel provided its approval to participate. As far as research settings, the hotel provides free open public Wi-Fi in their public spaces such as lobbies, restaurants, bars, and coffee shops. The conclusions reached could be extrapolated to other public Wi-Fi hotspots, as long as the design assumptions documented in this study apply. Generalization to other public hotspots may not be warranted. In addition, the client-side evil twin AP attack detection system built as part of this study is based on laptop platform with Linux OS. Generalization to other mobile platforms and operating systems may not be warranted.

## Sample Population

The sample population in this study consists of a hotel property located in Ecuador. The hotel property offers free open public Wi-Fi to wireless users in hotel public areas such as lobbies, restaurants, bars and coffee shops.

## Data Analysis

To assess the prototype system and effectively answer the research questions in this study, quantitative data was collected and analyzed using Hossen & Wenyuan's (2014) performance analysis approach:

1. Used the following standard metrics:
    a) *Accuracy*: indicates how accurately the system detects evil twin AP attacks.
    b) *Precision*: is the fraction of positively detected attacks to all positively detected attacks (correctly or incorrectly).

   c) *Recall*: is the fraction of positively detected attacks to all attacks that should be positively detected.

2. Used True positive (TP), True negative (TN), False positive (FP), False negative (FN) to calculate above standard metrics:
   a) *TP and TN*: represent correct classification
   b) *FP and FN*: represent incorrect classification
3. Used the following equations to calculate standard metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. Used diagrams including performance (%) for accuracy, precision and recall to depict performance results.
5. Answered the research questions based on the performance results. In addition, the data was gathered and analyzed with the intent of showing that the principles, processes, methods, and technologies used as well as the issues encountered during the evaluation apply to other hotel public Wi-Fi environments.

## Format for Presenting Results

The design, development, and implementation of the artifact conveyed in this study is presented in support of Hevner's guideline 7 through the creation and communication of a complete solution manual including design, procedures and specifications and is made available via ProQuest Dissertations database. The solution manual includes the following sections:

1. Physical network connectivity design
2. Logical prototype topology design diagram
3. Artifact construction specifications
4. Minimum hardware and software requirements
5. Step-by-step artifact construction procedures
6. Step-by-step artifact testing procedures
7. Transition client-side mobile evil twin attack detection system into production

### Resource Requirements

Scholarly and industry publications such as journal articles, textbooks, conference proceedings, technical reports, research reports, and online newspapers were used to support the client-side detection system. This section addressed the resources that were under the researcher’s control in order to complete the research:

- Hardware (Cisco router, Cisco switch, Cisco wireless controller, Cisco wireless access points, Lenovo laptops, Motorola Android smartphone);
- Software (Linux OS, Java SE Development Kit, Netbeans IDE, Wireshark packet analyzer, Kali Linux (Aircrack-ng), Hostapd, Android Mobile Hotspot and Tethering);
- Client-side evil twin attack detection system; and
- Access to free open public Wi-Fi at a hotel (public Wi-Fi spaces)

## Summary

Based on Hevner et al.'s (2004) seven guidelines of DSR, Peffers et al.'s DSRM (2008), and Hossen & Wenyuan's (2014) study evaluation method this chapter of the dissertation report delivered the structure for the development of design procedures and specifications derived from DSR literature to guide the construction, implementation, and evaluation of an effective client-side evil twin attack detection architecture artifact (CSMETAD). The research problem and the methodology of how to realize the desired outcome of building and evaluating the artifact to be used to support users of free open public Wi-Fi was achieved by delineating a two phased research approach. The first phase of the research emphasized the development of design principles, procedures and specifications that guided the artifact design, construction, implementation, and evaluation of the client-side evil twin attack detection artifact based on design science methodologies. The second phase of the research evaluated the effectiveness and efficiency of the artifact by implementing a prototype system.

# Chapter 4

# Results

This chapter presents the results of the activities associated with the system demonstration and evaluation phases described in previous chapter. The system demonstration phase includes lab deployment activities. The system evaluation phase includes the methods used to evaluate the artifact in the field, experiment results, followed by an analysis of the artifact's performance.

## System Demonstration

Lab deployment activities involved testing and evaluation of the client-side mobile evil twin attack detection system contained in a controlled environment as presented in chapter 3. The prototype system was tested and evaluated in the lab to analyze the requirements and demonstrate its effectiveness in a controlled environment. The lab simulated the hotel public Wi-Fi hotspot and provided an effective environment suitable for testing. Requirements were analyzed using observations and results from the lab experiment. Over the two-week duration of the lab deployment activities, the author used Hossen & Wenyuan's (2014) evaluation methodology to collect and analyze the data.

### *Requirements Analysis*

Several key requirements drove the research effort. These requirements were analyzed in the lab to demonstrate the artifact's effectiveness addressing the problem. The key requirements include:

*R1:* It protects users whether or not they have connected to a free open public Wi-Fi network in the past.

*R2:* It protects users when not all the hotspot APs with the desired SSID are detected during the initial wireless network scanning.

*R3:* It protects users when the client is not able to associate to all the APs in the public Wi-Fi network.

*R4:* It protects users when the attacker sets up the mobile evil twin AP with the same SSID, BSSID, and subnet of a legitimate AP.

*R5:* It protects users when the attacker blocks access to the public website used to get ISP information.

*R6:* It protects users when the attacker presents an invalid certificate while retrieving ISP information from a public website.

*R7:* After mobile evil twin AP attack detection, it connects users to a legitimate AP.

*R8:* It protects users for the duration of the public Wi-Fi connection, discovering and reporting on new mobile evil twin access points.

The following provides the test procedures for each requirement and the results:

*R1: It will protect users whether or not they have connected to a free open public Wi-Fi network in the past.*

This requirement was tested during phase 1 of the algorithm. To test this requirement, the wireless client was set up with a previous connection to the public Wi-Fi network, in this case the lab SSID (labwifi). The system was initialized by the user before visiting the open public Wi-Fi hotspot. This is required only the first time the system is used in the public Wi-Fi hotspot. Results show that the system iterated through all the 802.11

wireless network connections and validated if connection autoconnect was enabled and unencrypted.  After validation, the system disabled autoconnect for all open (unencrypted) 802.11 wireless network connections.

*R2:  It will protect the user when not all the hotspot APs with the desired SSID are detected during initial wireless network scanning.*

This requirement was tested during phase 2 of the algorithm.  In phase 2, after the system received a disassociated wireless card event, the system rescanned the wireless network to rediscover APs with selected SSID that were not detected during the initial wireless network scanning.  APs with signal strength equal to or greater than -75 dBm were added to the APs for selected SSID list.  Results show that when the number of APs for selected SSID was less than 1, the system presented the following message: “Your device is out of range for the selected public Wi-Fi hotspot.  Please move closer”.  The system then rescanned the public Wi-Fi network.

*R3: It will protect the user when the client is not able to associate to all the APs in the public Wi-Fi network.*

This requirement was tested during phase 1 and 2 of the algorithm.  In phase 1 and 2, the system attempted to associate to an AP with a timeout of 5 seconds.  The system checked association status each second for 5 seconds.  In phase 1, results show that when the system was not able to associate to an AP within 5 seconds, the system updated the APs for selected SSID list with association status as “false”, AP state as “unknown”, and client DHCP address, Internet access, secured public website access, certificate status, and global IP address as “not detected”.  Next, the system attempted to associate to the next AP in the APs for selected SSID list.  In phase 1, when the system was not able to associate to any of the APs, it presented the following message: “There is not enough information to categorize APs”, updated APs for selected SSID with AP state as “unknown”, and the system ended. In phase 2, results show that when the system was not able to associate to an AP within 5 seconds, it attempted to associate to the next AP with the highest signal strength in the APs for selected SSID list. In phase 2, when the system was not able to associate to any of the APs, it rescanned the public Wi-Fi network.

*R4 & R7 & R8: It will protect users when the attacker sets up the mobile evil twin AP with the same SSID, BSSID (MAC address), and subnet of a legitimate AP. After detection, it connects the user to a legitimate AP. Lastly, it protects the user for the duration of the public Wi-Fi connection, discovering and reporting on new mobile evil twin access points.*

Requirements 4, 7, and 8 were tested during phase 2 of the algorithm. To test these requirements, the system was run, collected data, detected only legitimate APs, connected the user to a legitimate AP, and waited for a disassociated wireless card event. No evil twin AP was present when the user ran the system. The author simulated the scenario when the attacker arrived later at the public Wi-Fi hotspot, configured the mobile evil twin AP with the same SSID, MAC address, and subnet of a legitimate AP, placed the mobile evil twin AP closer to the user (better signal strength), and proceeded to disassociate the user from the legitimate AP to force the user to connect to the mobile evil twin AP.

The system, after detecting a disassociated wireless card event, it rescanned the wireless network to rediscover APs with the same SSID that were not detected during the initial wireless network scanning.  In this scenario, the rescan showed two APs, one with signal strength equal to or greater than -75 dBm and one with signal strength less than -75 dBm.  The system added all APs for selected SSID with signal strength equal to or greater than -75 dBm to the APs for selected SSID list (new list).  This resulted in only one AP added.  Since there were two APs with the same SSID, MAC address, and subnet, the system only showed the AP that had the highest signal strength and ignored the other AP.  The system does not trust rescanned APs even if they have the same SSID, MAC address and subnet of legitimate APs; therefore, the system proceeds to validate them again.  Next, the system validated if the number of APs for selected SSID was equal to or greater than 1.  Results show that when the number of APs for selected SSID was equal to or greater than 1, the system retrieved learned ETA MAC addresses from learned ETA MAC address list and removed them from the APs for selected SSID list.  In this case, no ETA MAC addresses were retrieved and removed.  The system validated that the number of APs for selected was equal to or greater than 1.  Results show that when the number of APs for selected SSID was equal to or greater than 1, the system attempted to associate to the AP with the highest signal strength in the APs for selected SSID list.  When the system associated to the AP, obtained DHCP address, confirmed Internet access, accessed secured public website and verified that the public website certificate was valid, only then, the system obtained the global IP address.  When the AP global IP address was not the same as the trusted global IP address, the system printed access as “true”, certificate status as “valid”, and the global IP address of the mobile evil twin AP.  Next, the system presented the following message: “CSMETAD has detected an ETA on the public Wi-Fi hotspot”, added the AP MAC address to the Learned ETA MAC address list, disassociated from current AP and associated to the next AP with the highest signal strength in the APs for selected SSID list. Since there were no more APs in the list, the system displayed the following message: "CSMETAD finished iterating across the list of APs and was not able to validate that the AP global IP was the same as the trusted global IP" and proceeded to scan the public Wi-Fi network again.

The results of the rescan showed two APs, one with signal strength equal to or greater than -75 dBm and one with signal strength less than -75 dBm. The system added all APs for selected SSID with signal strength equal to or greater than -75 dBm to the APs for selected SSID list. In this case, only one AP with the same MAC address as previously learned ETA MAC address was added to the list. Same as above, since there were two APs with the same SSID, MAC address, and subnet, the system only showed the AP that had the highest signal strength and ignored the other AP. Next, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system retrieved learned ETA MAC addresses from learned ETA MAC address list and removed them from APs for selected SSID list, leaving no more APs in the list. The system then validated that the number of APs for selected was equal to or greater than 1. Results show that when the number of APs for selected SSID was less than 1, the system presented the following message: "You are located on the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi Hotspot" and proceeded to rescan the public Wi-Fi network.

The results of the rescan this time showed two APs, both with signal strength equal to or greater than -75 dBm. The system added all APs for selected SSID with signal strength equal to or greater than -75 dBm to the APs for selected SSID list. In this case, two APs were added to the list, one AP with the same MAC address as previously learned ETA MAC address and a second AP. Same as above, since there were two APs with the same SSID, MAC address, and subnet, the system only showed the AP that had the highest signal strength and ignored the other AP. Next, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system retrieved learned ETA MAC addresses from learned ETA MAC address list and removed them from APs for selected SSID list, leaving only one AP in the list. The system then validated that the number of APs for selected was equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system attempted to associate to the AP with the highest signal strength in the APs for selected SSID list. When the system associated to the next AP in the list and obtained DHCP address, confirmed Internet access, accessed secured public website and verified that the public website certificate was valid, only then, the system obtained the global IP address. When the AP global IP address was the same as the trusted global IP address, the system printed access as “true”, certificate status as “valid”, and the global IP address of the legitimate AP. Next, the system connected the user to the legitimate AP and presented the following message: “Wi-Fi connection is safe. You are connected to a legitimate AP”, and waited for a disassociated wireless card event. Results show that when the system received a disassociated wireless card event, the system rescanned the public Wi-Fi network and the algorithm phase 2 repeated (infinite loop).

When the system associated to the next AP in the list but was not able to obtain the DHCP address, confirm Internet access, access secured public website or verify that the public website certificate was valid, the system disassociated from AP and associated to the next AP in the APs for selected SSID list. When there were no more APs in the list, the system rescanned the public Wi-Fi network.

*R5: It protects the user when the attacker blocks access to the public website used to get ISP information.*

Requirement 5 was tested during phase 2 of the algorithm. To test this requirement, the author simulated the scenario when the attacker blocks access to the secured public website used to collect the global IP address using iptables rules.

From above example, when the system associated to an AP, obtained DHCP address, and confirmed Internet access, it proceeded to access secured public website to obtain the global IP. Results show that when the system was blocked access to the secured public website, it printed access as “false”, certificate status as “not detected” and global IP as “not detected”. Next, the system disassociated from AP and associated to the next AP in the APs for selected SSID list.

*R6: It protects the user when the attacker presents an invalid certificate while retrieving ISP information from a public website.*

To test an invalid public website certificate would require the creation of a fake remote server which will be very expensive to set up. This requirement was tested only in the lab. To test this requirement, two websites that provide invalid certificates were used to simulate the attacker using a fake remote server to bypass detection procedure (Google Open Source, n.d.). Results show that when the system verified that the public website certificate was invalid, the system printed access as “true”, certificate status as “invalid”, and global IP as “not detected”. The system then updated the AP MAC address list with AP state as “ETA”, added the AP MAC address to the Learned ETA MAC address list, disassociated from AP, and associated to the next AP in the APs for selected SSID list.

Appendix B shows Algorithm test cases and results. Appendix C shows CSMETAD system results for each of the key requirements.

### *Lab – Performance*

Performance metrics from Hossen & Wenyuan (2014) as depicted in chapter 3 were used to calculate the effectiveness of the artifact in a controlled environment. The findings showed that CSMETAD can detect mobile evil twin AP attacks effectively. CSMETAD detected mobile evil twin attacks, with 100% accuracy, precision, and recall. CSMETAD performance in detecting mobile evil twin attacks in the lab is depicted in Figure 13.

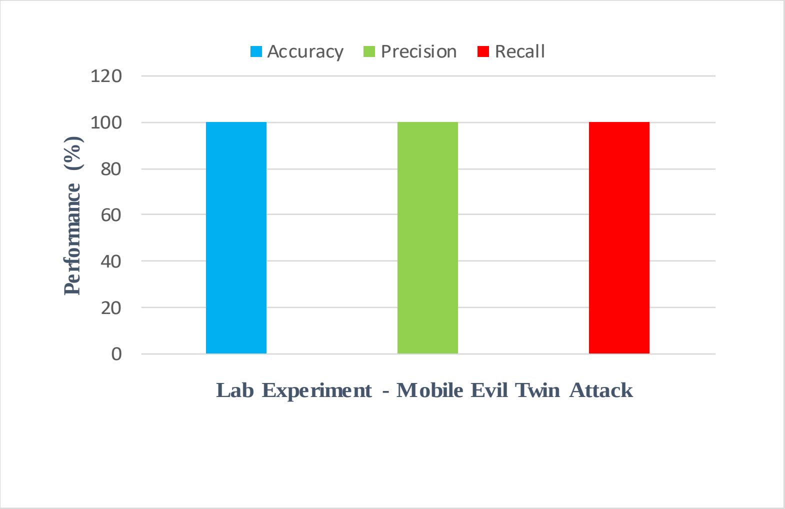


*Figure 13.* CSMETAD performance for mobile evil twin attacks in the lab.

## System Evaluation

At the conclusion of the lab deployment activities, the author travelled to Ecuador to extensively evaluate the robustness of the client-side mobile evil twin attack detection (CSMETAD) system in practice at a hotel public Wi-Fi hotspot that provides free open public Wi-Fi in its public spaces (restaurant, lobby, coffee shop and bar). Initially, the CSMETAD system aimed at searching for mobile evil twin attacks in the wild. Since no evil twin AP attacks were detected during the field evaluation period, the author proceeded to evaluate the system with the mobile evil twin AP used in the lab. Similar approach was used by Hossen & Wenyuan's study and the remainder of the client-side evil twin attack detection studies referenced in this dissertation. Over the three-week duration of the system evaluation, the author used Hossen & Wenyuan's (2014) evaluation methodology to collect and analyze the data. The client-side evil twin detection method was evaluated with real-world scenarios.

### *Observations*

During wild testing, the author ran the system in several locations in the hotel public Wi-Fi hotspot during busy times and waited for attackers to perform mobile evil twin attacks. During each run, the system collected data, connected the user to a legitimate AP and waited for a disassociated wireless card event caused by an attacker (refer to scenario 1 below). If no disassociations were detected after 10 to 30 minutes, the author re-ran the system. Results show that no mobile evil twin APs were detected during the field evaluation period.

Since no mobile evil twin attacks were detected in the wild, the author proceeded to evaluate the system with the mobile evil twin AP used in the lab. The author simulated the scenario of when the user ran the system, the system collected data, connected the user to a legitimate AP, and waited for a disassociated wireless card event. Attackers with mobile evil twin APs were not present when the user ran the system at the hotel public Wi-Fi hotspot. Next, the attacker with a mobile evil twin AP arrived at the public Wi-Fi hotspot, disassociated the user from the legitimate AP, and forced the user to connect to the mobile evil twin AP. The system then proceeded to detect mobile evil twin attacks, connect the user to a legitimate AP, and protect the user for the duration of the public Wi-Fi connection. In the tests, the public website certificate was valid. Detailed experimental results are described in the following paragraphs.

***Scenario 1: User ran the system, system collected data, connected the user to a legitimate AP, and waited for a disassociated wireless card event. Attacker with a mobile evil twin AP was not present when the user arrived to the public Wi-Fi hotspot and ran the system.***

*Scenario 1 - Basic Flow - Phase 1 – Data Collection*

The system at the beginning of phase 1 disabled auto-connections to all open (unencrypted) public Wi-Fi networks. The system then scanned the wireless network to discover SSIDs in range and presented them to the user. After the user selected the hotel public Wi-Fi hotspot SSID, the system created a list including all APs for selected SSID with signal strength equal to or greater than -75 dBm. The system detected two APs. The system then validated if the number of APs for selected SSID was equal to or greater than 2. Results show that when the number of APs for selected SSID was equal to or greater than 2, the system presented the following message: “There is sufficient information to start detecting ETAs”.

The system then started to iterate across all the APs and attempted to associate to each AP in the APs for selected SSID list. After successful AP association, the system proceeded to obtain client DHCP information, confirm Internet access, access secured public website and verify that the public website certificate was valid. Results show that when the system was able to associate to an AP, obtain client DHCP information, confirm Internet access, access secured public website and verify that the public website certificate was valid, only then, the system was able to obtain the global IP address. To verify AP association, the system displayed the elements of the network interface specific to the wireless operation. The system printed: “Client has associated to AP”. To verify DHCP information, the system displayed configured network interface parameters. The system printed: “Found IP address: xxx.xxx.xx.xx”. To verify Internet access, the system attempted to access an URL to check Internet connection. If the system was able to access the URL, then the system proceeded to approve terms of use. When redirection to a captive portal was detected, the system accepted terms of use and checked the Internet connection. The system printed: “CSMETAD was able to accept terms of use. Internet access confirmed”. When redirection to a captive portal was not detected, Internet access was confirmed. The system printed: “Internet access confirmed”. If the system was not able to access the URL, then the system printed: “Terms of Use approval failed”. To verify access to secured public website and that the public website certificate was valid, the system displayed cipher suite parameters and printed secured public website access as “true”, certificate status as “valid”, and global IP address results.

Once the system finished iterating throughout all the APs, the system determined the trusted global IP address to be used by the system for the duration of the public Wi-Fi connection by validating the number of global IP address occurrences. Results show that when the number of global IP address occurrences was equal to or greater than 2, the system assigned the global IP address as the trusted global IP and proceeded to iterate throughout all APs to categorize them as either “valid”, “ETA”, or “unknown”. In this scenario, the AP global IP addresses in the list were the same as the trusted global IP address; therefore, the system updated the APs for selected SSID list with AP state as “valid”.

*Scenario 1 - Alternative Flows - Phase 1 – Data Collection*

During the initial phase 1 scan, results show that when the number of APs for selected SSID was less than 2, the system presented the following message: “There is insufficient information to detect ETAs”, and the system ended.

During AP association, results show that when the system was not able to associate to an AP in the first attempt, the system slept for one second and re-attempted association. The system attempted to associate to an AP for 5 seconds. When the system was not able to associate to an AP in 5 seconds, it updated the AP state as “unknown” and attempted to associate to the next AP in the APs for selected SSID list.

After successful AP association, results show that when the system was not able to obtain DHCP information, confirm Internet access, access secured public website or verify that the public website certificate was valid, the system updated the AP state as “unknown”, disassociated from current AP and associated to the next AP in the APs for selected SSID list.

During the determination of the trusted global IP, results show that when the number of global IP address occurrences was less than 2, the system displayed the following message: “There is not enough information to categorize APs”, updated AP state as “unknown”, and the system ended.

*Scenario 1 - Basic Flow - Phase 2 – Detection and Protection*

In phase 2, the system proceeded to rescan the public Wi-Fi network to rediscover APs with selected SSID. The system added all APs for selected SSID with signal strength equal to or greater than -75 dBm to the APs for selected SSID list (new list). The system detected two APs with the same MAC addresses as phase 1. Next, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system retrieved learned ETA MAC addresses from learned ETA MAC addresses list and removed learned ETA MAC addresses from APs for selected SSID list. In the tests, since no mobile evil twin APs were detected in phase 1, no ETA MAC addresses were retrieved and removed. The system then validated if the number of APs for selected SSID continues to be equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system started to iterate across all the APs. First, the system attempted to associate to the AP with the highest signal strength in the APs for selected SSID list. After successful AP association, the system proceeded to obtain client DHCP information, confirm Internet access, access secured public website and verify that the public website certificate was valid. Results show that when the system was able to associate to an AP, obtain client DHCP information, confirm Internet access, access secured public website and verify the public website certificate was valid, only then, the system obtained the global IP address. To verify AP association, the system displayed the elements of the network interface specific to the wireless operation. The system printed: “Client has associated to AP”. To verify DHCP information, the system displayed configured network interface parameters. The system printed: “Found IP address: xxx.xxx.xx.xx”. To verify Internet access, the system attempted to access an URL to check Internet connection. If the system was able to access the URL, then the system proceeded to approve terms of use. When redirection to a captive portal was detected, the system accepted terms of use and checked the Internet connection. The system printed: “CSMETAD was able to accept terms of use. Internet access confirmed”. When redirection to a captive portal was not detected, Internet access was confirmed. The system printed: “Internet access confirmed”. If the system was not able to access the URL, then the system printed: “Terms of Use approval failed”. To verify access to secured public website and that the public website certificate was valid, the system displayed cipher suite parameters and printed secured public website access as “true”, certificate status as “valid”, and global IP address results.

The system then validated if the AP global IP address was the same as the trusted global IP address. Results show that the AP global IP address was the same as the trusted global IP address, the system connected to the AP and presented the following message: “Wi-Fi connection is safe. You are connected to a legitimate AP”. The system then waited for a disassociated wireless card event.

### *Scenario 1 - Alternative Flow – Phase 2 – Detection and Protection*

During the rescan, results show that when the number of APs for selected SSID was less than 1, the system presented the following message: "Your device is out of range for the selected public Wi-Fi hotspot. Please move closer". The system then rescanned the public Wi-Fi network.

After retrieving the learned ETA MAC addresses from the learned ETA MAC addresses list and removing learned ETA MAC addresses from APs for selected SSID list, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was less than 1, the system presented the following message: "You are located in the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi hotspot". The system then rescanned the public Wi-Fi network.

During AP association, results show that when the system was not able to associate to an AP in the first attempt, the system slept for 1 second and re-attempted association. The system attempted to associate to an AP for 5 seconds. When the system was not able to associate to an AP in 5 seconds, it attempted to associate to the next AP in the APs for selected SSID list. When the system was not able to associate to any APs, it rescanned the public Wi-Fi network.

After successful AP association, results show that when the system was not able to obtain DHCP information, confirm Internet access, access to secured public website or verify that the public website certificate was valid, the system disassociated from current AP and associated to the next AP in the APs for selected SSID list.

***Scenario 2: User was connected to a legitimate AP. Attacker arrived at the public Wi-Fi hotspot with a mobile evil twin AP, disassociated the user from legitimate AP, and forced the user to connect to the mobile evil twin AP (higher signal strength). Attacker set up the mobile evil twin AP with the same SSID, MAC address, and subnet as the legitimate AP.***

*Scenario 2 - Basic Flow - Phase 2 – Detection and Protection*

The system received a disassociated wireless card event caused by an attacker and proceeded to rescan the public Wi-Fi network to rediscover APs with selected SSID.

*Case 1: System detected two APs (One AP with signal strength equal to or greater than -75 dBm and one AP with signal strength less than -75 dBm).*

The system added all APs for selected SSID with signal strength equal to or greater than -75 dBm to the APs for selected SSID list. This resulted in only one AP added to the list. This AP had the same SSID, MAC address, and subnet of a legitimate AP. Next, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system retrieved learned ETA MAC addresses from learned ETA MAC addresses list and removed learned ETA MAC addresses from APs for selected SSID list. Since no mobile evil twin AP was detected in scenario 1, no ETA MAC addresses were retrieved and removed. The system then validated if the number of APs for selected SSID continues to be equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system started to iterate across APs. First, the system attempted to associate to the AP with the highest signal strength in the APs for selected SSID list. After successful AP association, the system proceeded to obtain client DHCP information, confirm Internet access, access secured public website and verify that the public website certificate was valid. Results show that when the system was able to associate to an AP, obtain the client DHCP information, confirm Internet access, access secured public website and verify that public website certificate was valid, only then, the system obtained the global IP address. To verify AP association, the system displayed the elements of the network interface specific to the wireless operation. The system printed: “Client has associated to AP”. To verify DHCP information, the system displayed configured network interface parameters. The system printed: “Found IP address: xxx.xxx.xx.xx”. To verify Internet access, the system attempted to access an URL to check Internet connection. If the system was able to access the URL, then the system proceeded to approve terms of use. When redirection to a captive portal was detected, the system accepted terms of use and checked the Internet connection. The system printed: “CSMETAD was able to accept terms of use. Internet access confirmed”. When redirection to a captive portal was not detected, Internet access was confirmed. The system printed: “Internet access confirmed”. If the system was not able to access the URL, then the system printed: “Terms of Use approval failed”. To verify access to secured public website and that the public website certificate was valid, the system displayed cipher suite parameters and printed secured public website access as “true”, certificate status as “valid”, and global IP address results.

The system then validated if the AP global IP address was the same as the trusted global IP address. Results show that the AP global IP address was not the same as the trusted global IP address, the system detected an ETA, and presented the following message: “CSMETAD has detected an ETA on the public Wi-Fi hotspot”, added the AP

MAC address to the learned ETA MAC address list, disassociated from current AP and associated to the next AP in the APs for selected SSID list.

Since there were no more APs in the list, the system displayed the following message: “CSMETAD finished iterating across the list of APs and was not able to validate that the AP global IP was the same as the trusted global IP” and proceeded to rescan the public Wi-Fi network.

*Case 2: System detected two APs (One of the APs had the same MAC address as previously detected ETA. Both APs with signal strength equal to or greater than -75 dBm).*

The system added all APs for selected SSID with signal strength equal to or greater than -75 dBm to the APs for selected SSID list. Next, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system retrieved learned ETA MAC addresses from learned ETA MAC addresses list and removed learned ETA MAC addresses from APs for selected SSID list. In this case, the AP with the same MAC address as previously detected ETA was removed from the APs for selected SSID list. After removing learned ETA MAC addresses from APs for selected SSID list, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system started to iterate across APs. The system associated with the AP with the highest signal strength. After successful AP association, the system proceeded to obtain client DHCP information, confirm Internet access, access secured public website and verify that the public website certificate was valid. Results show that when the system was able to associate to an AP, obtain the client DHCP information, confirm Internet access, access secured public website and validate public website certificate was valid, only then, the system obtained the global IP address. To verify AP association, the system displayed the elements of the network interface specific to the wireless operation. The system printed: "Client has associated to AP". To verify DHCP information, the system displayed configured network interface parameters. The system printed: "Found IP address: xxx.xxx.xx.xx". To verify Internet access, the system attempted to access an URL to check Internet connection. If the system was able to access the URL, then the system proceeded to approve terms of use. When redirection to a captive portal was detected, the system accepted terms of use and checked the Internet connection. The system printed: "CSMETAD was able to accept terms of use. Internet access confirmed". When redirection to a captive portal was not detected, Internet access was confirmed. The system printed: "Internet access confirmed". If the system was not able to access the URL, then the system printed: "Terms of Use approval failed". To verify access to secured public website and that the public website certificate was valid, the system displayed cipher suite parameters and printed secured public website access as "true", certificate status as "valid", and global IP address results.

The system then validated if the AP global IP address was the same as the trusted global IP address. Results show that the AP global IP address was the same as the trusted global IP address, the system connected to the AP and presented the following message: "Wi-Fi connection is safe. You are connected to a legitimate AP". Results show that when the system connected to a legitimate AP, it then waited for a disassociated wireless card event. When the system received a disassociated wireless card event, it proceeded to rescan the public Wi-Fi network protecting the user for the duration of the public Wi-Fi connection discovering and reporting on new mobile evil twin APs. Algorithm phase 2 repeated (infinite loop).

*Case 3: System detected two APs (One AP with the same MAC address as the previously detected ETA and signal strength greater than -75 dBm and one AP with signal strength less than -75 dBm).*

The system added all APs for selected SSID with signal strength equal to or greater than -75 dBm to the APs for selected SSID list. Only the AP with the same MAC address as the previously detected ETA was added to the list. Next, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was equal to or greater than 1, the system retrieved learned ETA MAC addresses from learned ETA MAC addresses list and removed learned ETA MAC addresses from APs for selected SSID list. In this case, the AP with the same MAC address as previously detected ETA was removed from the APs for selected SSID list, leaving no more APs in the list. After removing learned ETA MAC addresses from APs for selected SSID list, the system validated if the number of APs for selected SSID was equal to or greater than 1. Results show that when the number of APs for selected SSID was less than 1, the system presented the following message: “You are located in the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi hotspot”. The system then rescanned the public Wi-Fi network.

### *Scenario 2 - Alternative Flow – Phase 2 – Detection and Protection*

After the first rescan, results show that when the number of APs for selected SSID was less than 1, the system presented the following message: “Your device is out of range for the selected public Wi-Fi hotspot.  Please move closer”.  The system then rescanned the public Wi-Fi network.

During AP association, results show that when the system was not able to associate to an AP in the first attempt, the system slept for 1 second and re-attempted association. The system attempted to associate to an AP for 5 seconds.  When the system was not able to associate to an AP in 5 seconds, it attempted to associate to the next AP in the APs for selected SSID list.  When the system was not able to associate to any APs, it rescanned the public Wi-Fi network.

After successful AP association, results also show that when the system was not able to obtain DHCP information, confirm Internet access, access secured public website or verify public website certificate was valid, the system disassociated from current AP and associated to the next AP with the highest signal strength in the APs for selected SSID list.

*Field – Performance*

Performance metrics from Hossen & Wenyuan (2014) as presented in chapter 3 were used to measure the effectiveness of the artifact in an uncontrolled environment. CSMETAD detected mobile evil twin attacks, with 100% accuracy, precision, and recall. CSMETAD performance in detecting mobile evil twin attacks is depicted in Figure 14.

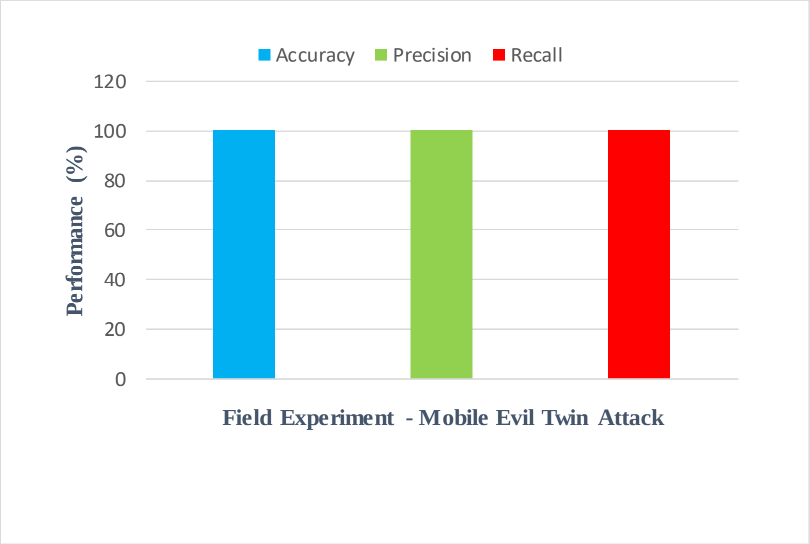


*Figure 14.* CSMETAD performance for mobile evil twin attacks in the field.

*Field – Time Delay Analysis*

Analysis of time delay was conducted using field data. Delay mainly consisted of time to associate to APs, collect DHCP information, confirm Internet access, connect to secured public website, verify that the public website certificate was valid, receive response from webserver, and connect the user to a legitimate AP after detection of mobile evil twin AP. The author measured the time delay for 3 main steps in the detection algorithm. The test was repeated 50 times.

a. **Time to associate to an AP and obtain a valid IP address from the DHCP server.** The results show that the average time to associate to an AP was 1 second and the average time to obtain a valid IP address from the DHCP server was 2 seconds. Total of 3 seconds for both parameters. Many factors affect these values, such as wireless network devices, wireless network's connection, and DHCP server.

b. **Time to confirm Internet access.** The results show that the average to confirm Internet access was 0.5 seconds. These time values depend on factors such as captive portal and Internet speed.

c. **Time to connect to secured public website, verify that the public website certificate was valid, and receive a response from the webserver.** The results show that the average duration of time required to connect to the secured public website, verify that the public website certificate was valid, and receive a response from the webserver was 0.7 seconds. Many factors affect these values, such as Internet speed, DNS response time, and webserver's response time.

For the three AP scenario (two legitimate APs and one mobile evil twin AP), data collection was completed within 8.2 seconds, mobile evil twin AP detection took approximately 5.2 seconds and the connection to a legitimate AP after detection was completed in 3.8 seconds.

Figure 15, 16 and 17 illustrate the results of the measurements.

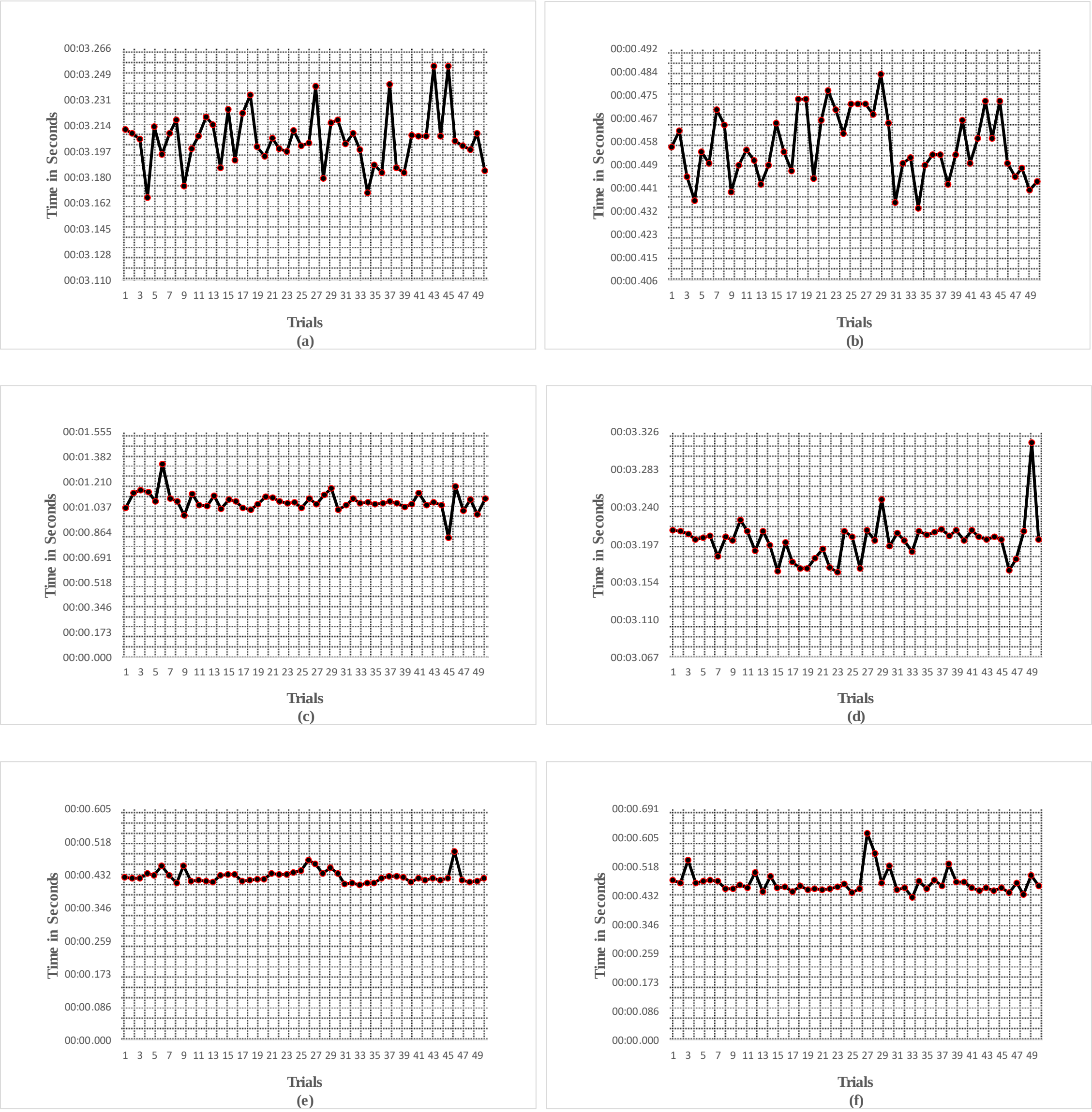


*Figure 15.* Time delay - Data Collection. a) connecting to legitimate AP1. (b) confirming Internet access. (c) connecting to public website, verifying certificate, and receiving a response from the webserver. d) connecting to legitimate AP2. (e) confirming Internet access. (f) connecting to public website, verifying certificate, and receiving a response from the webserver.

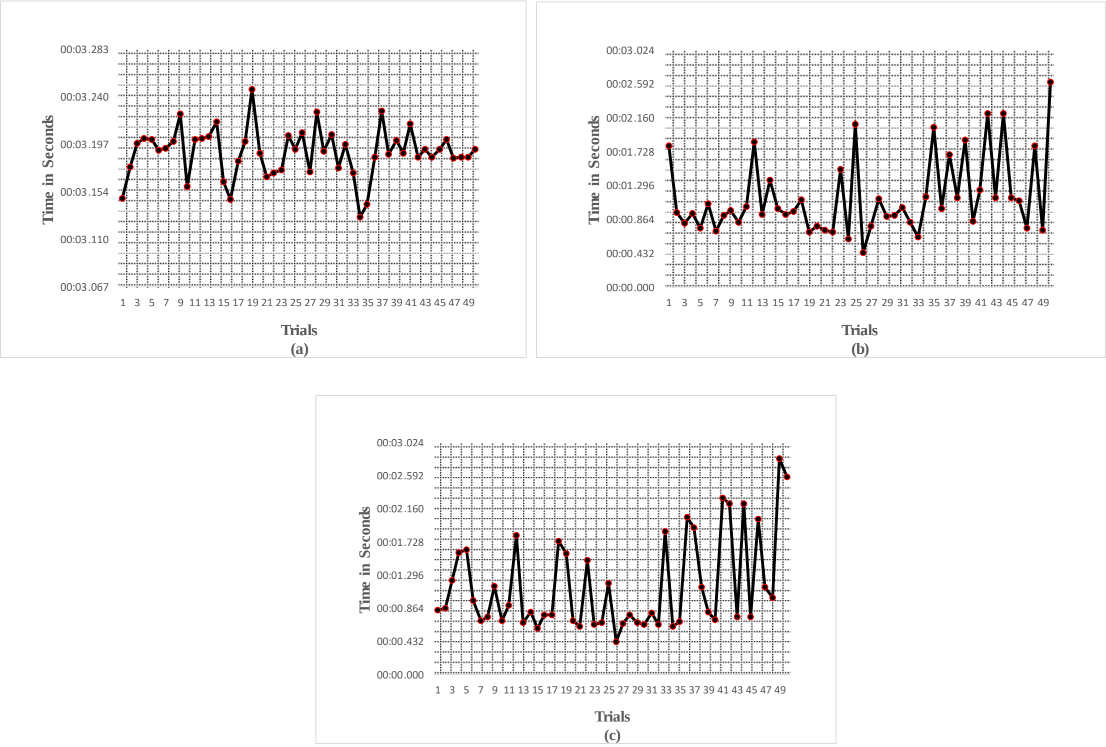


*Figure 16.* Time delay - Mobile Evil Twin AP Detection. a) connecting to mobile evil twin AP. (b) confirming Internet access. (c) connecting to public website, verifying certificate, and receiving a response from the webserver.

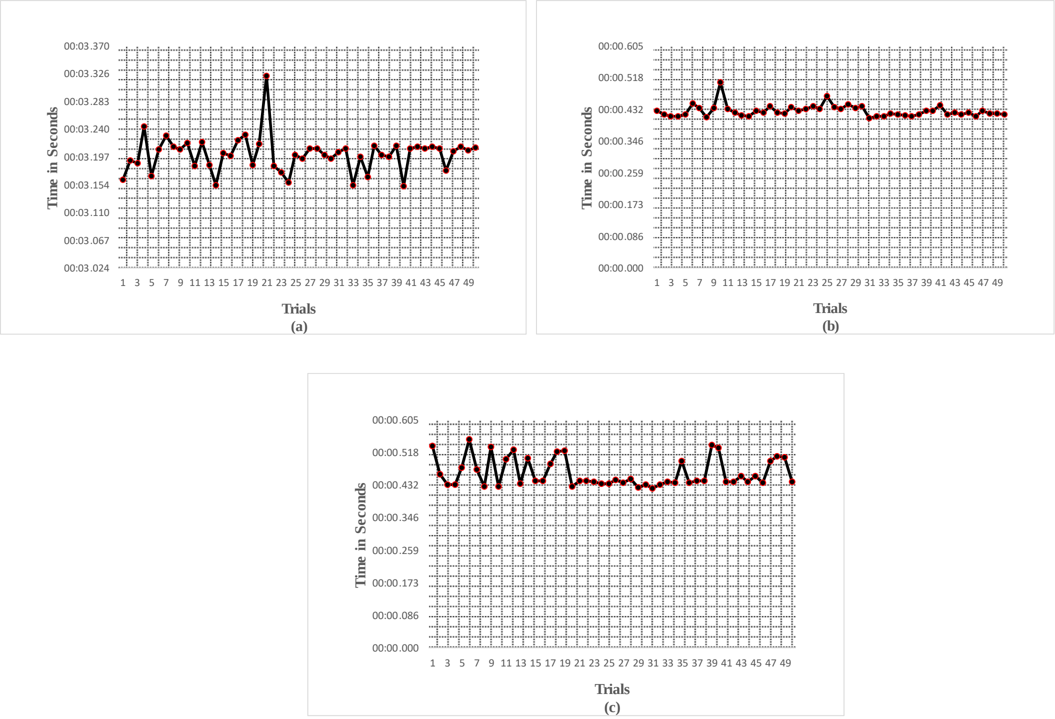


*Figure 17.* Time delay – Connection to legitimate AP after detection. a) connecting to legitimate AP. (b) confirming Internet access. (c) connecting to public website, verifying certificate, and receiving a response from the webserver.

## Summary of Results

As a result of this research and data analysis findings, experimental results show that CSMETAD can effectively detect and protect users from mobile evil twin attacks in public Wi-Fi hotspots with 100% accuracy, precision and recall. Data collection was completed within 8.2 seconds, mobile evil twin AP detection took approximately 5.2 seconds and the connection to a legitimate AP after detection was completed in 3.8 seconds. Although, time delay may vary according to many factors as explained above, these factors did not affect the detection effectiveness.

# Chapter 5

# Conclusions, Implications, and Recommendations

This chapter presents the conclusions of this study, including its strengths, weakness, and limitations. The chapter also includes the implications for actions and recommendations for future research. This chapter concludes with a summary of the study.

## Conclusions

For this investigation, the author sought to develop a more effective, efficient, and practical client-side evil twin attack detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi hotspots. To this end, this was an experimental study that used Hevner et al.'s (2004) seven guidelines of DSR, Peffers et al.'s DSRM (2008), and Hossen & Wenyuan's (2014) study evaluation methodology. The client-side evil twin attack detection system was validated by conducting a three-week field study at a hotel public Wi-Fi hotspot and tested against Hossen & Wenyuan's (2014) method for detecting mobile evil twin attacks.

Based on the analysis performed and the results achieved as presented in chapter 4, the specifics objectives of the research questions in this study haven been met based on evidence that is presented in the following pages and paragraphs.

The first research question asked for the requirements that the artifact must meet in order to address the problem. The answer to the first research question is provided in the form of requirements based on a thorough review of existing client-side evil twin attack detection literature addressing limitations regarding requirements, assumptions, and evaluation approaches as presented in chapter 3.

The second research question asked for the major decision points in the design and development process.  The answer to this question is provided in the form of design principles, procedures and specifications based on DSR that supported the artifact construction, implementation, and evaluation of the client-side evil twin attack detection system as presented in chapter 3.

The third research question asked for the way the product developed meet or fail to meet the requirements specified.  The answer to this question is provided in the form of observations and results from the lab and field tested against Hossen and Wenyuan's detection method.  Details are presented below:

*R1:  It will protect users whether or not they have connected to a free open public Wi-Fi network in the past.*

Experimental results indicate that the system successfully met this requirement by disabling auto-connections to all open (unencrypted) public Wi-Fi connections, protecting the user from automatically connecting to a previously connected AP (potentially a mobile evil twin AP) when using the public Wi-Fi hotspot.

Hossen and Wenyuan's method did not meet this requirement since it does not protect users who have connected to an open (unencrypted) public Wi-Fi network in the past. Hossen & Wenyuan's method does not cover this scenario.

*R2: It will protect the user when not all the hotspot APs with the desired SSID are detected during initial wireless network scanning.*

Experimental results indicate that system successfully met this requirement. After the system received a disassociated wireless card event, the system rescanned the wireless network to rediscover APs with selected SSID that were not detected during the initial wireless network scanning. If after the rescan, the system did not detect any APs, the system rescanned the public Wi-Fi network. This approach allows for the system to work effectively.

Hossen and Wenyuan’s method did not meet this requirement since it assumes that all the public Wi-Fi APs are detected in the initial wireless network scanning, which in practice is not always the case. Hossen and Wenyuan’s method does not cover the scenario when not all the hotspot APs with the desired SSID are detected during the initial wireless network scanning.

*R3: It will protect the user when the client is not able to associate to all the APs in the public Wi-Fi network.*

Experimental results indicate that the system successfully met this requirement. The system checked association to an AP with a timeout of 5 seconds. In cases when the system was not able to associate to an AP within 5 seconds, it attempted to associate to the next AP in the APs for selected SSID list. In phase 1, when the system was not able to associate to any of the APs, the system updated AP state as “unknown” and the system ended. In phase 2, when the system was not able to associate to any of the APs, the system rescanned the public Wi-Fi network. This approach allowed for the system to be practical and effective.

Hossen and Wenyuan's method did not meet this requirement since it assumes that the client is able to associate to all the APs in the public Wi-Fi network, which in practice is not always the case. Hossen and Wenyuan's method does not cover the scenario when the client is not able to associate to all the APs in the public Wi-Fi network.

*R4: It will protect users when the attacker sets up the mobile evil twin AP with the same SSID, BSSID (MAC address), and subnet of a legitimate AP.*

Experimental results indicate that the system successfully met this requirement by detecting disassociated wireless network events caused by an attacker, rescanning the public Wi-Fi hotspot to rediscover APs with selected SSID, retrieving learned ETA MAC address from learned ETA MAC addresses list and removing learned ETA MAC addresses from APs for selected SSID list, validating APs even if they have the same SSID, MAC address and subnet as a legitimate AP, verifying AP association, DHCP information, Internet access, access to secured public website and public website certificate, to be able to get the global IP address and compare it with trusted global IP address. This approach allowed for the system to be effective.

Hossen and Wenyuan's method did not meet this requirement since it only protects users from mobile evil twin APs that have the same SSID. Method assumes that the AP MAC addresses are unique and use that as a reference to switch between APs. Also, Hossen & Wenyuan's method assumes that the mobile evil twin AP is in a different subnet as the legitimate AP. To avoid detection, the attacker could set up the mobile evil twin AP with the same SSID, MAC address, and subnet of a legitimate AP. Also, method assumes that the attacker is already in the hotspot and is connected when the wireless user runs the detection system. In a real life environment, an attacker may not be present when the user connects to the hotspot. Hossen & Wenyuan's method does not cover the scenario when the attacker sets up the mobile evil twin AP with the same SSID, MAC address, and subnet of a legitimate AP and when the attacker arrives at the public Wi-Fi hotspot at a later time. Additionally, Hossen & Wenyuan assume that the client is always able to associate to an AP, obtain DHCP address, confirm Internet access, and access secured public website, which in practice is not always the case.

*R5: It will protect users when the attacker blocks access to the public website used to get ISP information.*

Experimental results indicate that the system successfully met this requirement by detecting public website access blocking, disassociating from AP and associating to the next AP in the APs for selected SSID list.

Hossen and Wenyuan's method did not meet this requirement since it does not protect the user when the attacker blocks access to the public website used to get ISP information. An attacker who is aware of the algorithm would try to block the user access to the secured public website used to get ISP information. If the attacker blocks access to the website, the method will not work.

*R6: It will protect users when the attacker presents an invalid certificate while retrieving ISP information from a public website.*

This requirement was only simulated and tested in the lab. Lab results indicate that the system successfully met this requirement by verifying public website certificates presented by an attacker when the system access secured public website to retrieve ISP information; and if invalid, adding AP MAC address to learned ETA MAC address list, disassociating from AP and associating to the next AP from the APs for selected SSID list.

Hossen and Wenyuan’s method did not meet this requirement since it does not protect the user when the attacker presents an invalid certificate while retrieving ISP information from a public website. Method does not verify the public web server certificate. An attacker who is aware of the algorithm would try to present an invalid certificate while the system is retrieving ISP information. The attacker would create a fake remote server to bypass detection procedure. If the attacker presents an invalid certificate, the method will not work.

*R7: It will, after mobile evil twin AP detection, connect the user to a legitimate AP.*

Experimental results indicate that the system successfully met this requirement by validating remaining APs after mobile evil twin AP detection, associating to the next AP with the highest signal strength, verifying AP association, DHCP information, Internet access, access to secured public website and public website certificate, to be able to get the global IP address and compare it with trusted global IP address. This approach allows for seamless and secured public Wi-Fi experience in a public Wi-Fi location.

Hossen and Wenyuan’s method did not meet this requirement since after mobile evil twin AP detection, it does not connect the user to a legitimate AP. Method only warns the user of the presence of an evil twin AP. Method does not cover the scenario when after detection, it connects the user to a legitimate AP.

*R8: It will protect the user for the duration of the public Wi-Fi connection, discovering and reporting on new mobile evil twin access points.*

Experimental results indicate that the system successfully met this requirement by detecting a disassociated wireless card event and rescanning the public Wi-Fi network environment to rediscover and report on new mobile evil twin access points for the duration of the public Wi-Fi connection. Algorithm phase 2 repeated (infinite loop). The infinite loop approach allowed for the system to be practical and effective.

Hossen and Wenyuan’s method did not meet this requirement since it does not protect the public Wi-Fi users for the duration of the Wi-Fi connection, discovering and reporting on new mobile evil twin access points. Method assumes that the attacker is already in the hotspot and is connected when the wireless user connects to the public Wi-Fi network. In real life environment, an attacker may not be present when the user connects to the hotspot. Method does not address the case where the attacker arrives later at the hotspot.

In regards to system performance, results show that CSMETAD can effectively detect and protect users from mobile evil twin AP attacks in public Wi-Fi hotspots in various real-world scenarios despite time delay caused by many factors. Time delay details are provided separately for data collection, detection, and connection to legitimate AP to be used as a baseline for future studies.

### *Strengths*

The major strength of this investigation is that the system was designed and developed based on a thorough review of the existing client-side evil twin attack detection solutions literature and addressed limitations regarding requirements, assumptions, and evaluation approaches. Additionally, the DSR principles, procedures and specifications that supported the construction, implementation, and evaluation of the client-side evil twin attack detection system provided an approach to conducting field studies of a similar nature with focus on multiple real-world scenarios.

### *Weakness*

One weakness of this study is that the detection system built as part of this study does not operate under the assumption that the attacker performs an evil twin attack using the legitimate AP's Internet access. However, combining the detection method with other methods that were used to detect evil twin attacks using the legitimate AP's Internet access, such as the ones referenced in this dissertation, will provide a complete evil twin attack detection system.

### *Limitations*

One of the limitations in this study is that since the artifact was initially evaluated in the wild, the author was not able to control when attackers were going to appear at the hotel public Wi-Fi hotspot to perform mobile evil twin AP attacks. Since no evil twin APs were detected during the field evaluation period, the author proceeded to test the system using the lab mobile evil twin AP. Similar evaluation technique was used in Hossen & Wenyuan's and the remainder of client-side evil twin attack detection studies referenced in this dissertation.

Another limitation in this investigation was evaluation costs that prevented the author from evaluating the detection system using public Wi-Fi users (as users of the detection system) and at a large scale. In order to analyze and evaluate the artifact using public Wi-Fi users and at a large scale, the system would need to be made available to a large number of actual users who can test the system in many public Wi-Fi locations for a defined period and report back to the author on detection effectiveness and efficiency. This will improve the likelihood of detecting real mobile evil twin APs in the wild. Despite evaluation cost limitations, the scope of the study was appropriate and consistent with the budget.

## Implications

### *Impact on the Field of Study*

The goal of this study was to contribute to the body of knowledge of wireless security research by developing a more effective, efficient, and practical client-side evil twin attack detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi. The artifact was designed, developed and evaluated based on a thorough review of the existing client-side evil twin attack detection solutions literature and addressed limitations regarding requirements, assumptions, and evaluation approaches. Based on design science research (DSR) literature, Hevner's seven guidelines of DSR, Peffer's design science research methodology (DSRM), and Hossen & Wenyuan's (2014) study evaluation methodology, the author developed design principles, procedures and specifications to guide the construction, implementation, and evaluation of a prototype client-side evil twin attack detection artifact. The author evaluated the client-side evil twin attack detection system in a hotel public Wi-Fi environment. To the best of the author's knowledge, this is the first academic study in this field that attempts to detect mobile evil twin APs in the wild extensively at a public Wi-Fi hotspot. Since no evil twin APs were detected during the field evaluation period, the author proceeded to test the system with the mobile evil twin AP used in the lab. Adoption of this artifact by others will provide a detection method that can be improved to include additional real-world scenarios in studies of a similar nature.

*Implications for Future Research*

Implications for future research include a large-scale evaluation with real traveling users of the system in many public Wi-Fi hotspot locations.  Because this study focused on detecting mobile evil twin APs in a single public Wi-Fi hotspot, similar studies can be conducted in many public Wi-Fi hotspots improving the likelihood of detecting real mobile evil twin APs.  Conducting such studies would require improvements to the client-side mobile evil twin attack detection system.  Suggested improvements are presented in the section below.

## Recommendations

The system limitations observed by the author during this investigation primarily involved: (1) initializing the system before arriving to the hotspot to disable autoconnection to previously connected open public Wi-Fi hotspots; (2) waiting for a disassociated wireless card event to rescan the wireless network and connect to an AP with better signal strength when the signal level is below a threshold; (3) costs of creating a fake remote server to test the validity of a public website certificate; and (4) creating a user interface for a large-scale evaluation in the wild.  To address these limitations, the author recommends the following: (1) disabling autoconnect when the user installs the system; (2) rescanning the wireless network when the signal strength of an AP is below a determined threshold; (3) testing SSL exceptions with a larger set of invalid certificates; and (4) creating an effective graphical user interface.  Adoption of these recommendations could provide overall improvements to the client-side mobile evil twin attack detection system and facilitate broader adoption in similar studies.

To address the limitation of initializing the system before visiting the hotspot to disable autoconnection to previously connected open Wi-Fi connections, the author recommends disabling autoconnect when the user installs the system on his or her laptop. This approach would consist of a subset application that when installed and run will disable all previous open Wi-Fi connections.

Rescanning the wireless network when the signal strength of an AP is below a determined threshold would facilitate not waiting for a disassociated wireless card event to rescan the wireless network to connect to an AP that offers a better signal strength.

To address the limitation of costs of creating a fake remote server to test the validity of a public website certificate, the author recommends instead expanding the simulation approach used in this dissertation report to include testing of SSL exceptions with a larger list of invalid certificates.  An example of a website that includes a list on invalid SSL certificates is badssl.com (Google Open Source, n.d.).

To address creating a user interface for a large scale evaluation in the wild, the author recommends the development of a graphical user interface that displays simple user messages communicating detection and protection results.  The graphical user interface should be designed to facilitate a wide adoption and usability of the system by non-technical users.

## Summary

Users and providers benefit considerably from public Wi-Fi hotspots. Users receive wireless Internet access and providers draw new prospective customers. While users are able to enjoy the ease of Wi-Fi Internet hotspot networks in public more conveniently, they are more susceptible to a particular type of fraud and identify theft, referred to as evil twin attack (ETA). Through setting up an ETA, an attacker can intercept sensitive data such as passwords or credit card information by snooping into the communication links. Since the objective of free open (unencrypted) public Wi-Fi hotspots is to provide ease of accessibility and to entice customers, no security mechanisms are in place. The public’s lack of awareness of the security threat posed by free open public Wi-Fi hotspots makes this problem even more heinous. Client-side systems to help wireless users detect and protect themselves from evil twin attacks in public Wi-Fi hotspots are in great need.

The author explored the problem of the need for client-side detection systems that will allow wireless users to help protect their data from evil twin attacks while using free open public Wi-Fi. The client-side evil twin attack detection system developed as part of this dissertation linked the gap between the need for wireless security in free open public Wi-Fi hotspots and limitations in existing client-side evil twin attack detection solutions. The goal of this research was to develop a more effective, efficient, and practical client-side detection system for wireless users to independently detect and protect themselves from mobile evil twin attacks while using free open public Wi-Fi hotspots.

To address the research problem and the methodology of how to accomplish the stated goal of designing and building a more effective, efficient, and practical client-side artifact to be used to detect mobile evil twin attacks, the author utilized a two phased research approach. In phase one, the author developed design principles, procedures and specifications to guide the design, construction, implementation, and evaluation of the prototype client-side evil twin detection artifact using Hevner's seven guidelines of DSR, Peffer's design science research methodology (DSRM), and Hossen & Wenyuan's (2014) study evaluation methodology. In phase two, the author extensively evaluated the performance of the client-side evil twin attack detection method by implementing a prototype system. The prototype system was implemented and evaluated in two environments. First, in a lab to analyze the requirements and demonstrate the effectiveness in a controlled environment. Second, in the field at a hotel public Wi-Fi hotspot to extensively evaluate the robustness of the system in practice. The prototype system aimed at detecting real mobile evil twin APs in the wild at a hotel property that provides free open public Wi-Fi in its public spaces. Since no real evil twin APs were detected during the field evaluation period, the author proceeded to evaluate the system with the mobile evil twin AP used in the lab. Similar approach was used by Hossen & Wenyuan's study and the remainder of the client-side evil twin attack detection studies referenced in this dissertation.

The techniques to evaluate the effectiveness of the system were based on Hossen & Wenyuan's (2014) evaluation methodology which was published and validated. The client-side evil twin detection method developed as part of this dissertation was tested against Hossen & Wenyuan's (2014) method for detecting mobile evil twin attacks. The experimental results show that the CSMETAD system can effectively detect and protect users from mobile evil twin AP attacks in public Wi-Fi hotspots in various real-world scenarios despite time delay caused by many factors.

At the conclusion of this study, the author addressed the observed limitations of the study, discussed the implications for further research, and presented recommendations. The major limitations of the study were being able to detect mobile evil twin APs in the wild and evaluation costs. An implication for further research includes a large-scale evaluation with real traveling users of the system in many public Wi-Fi hotspot locations. Conducting such study would require improvements to the client-side mobile evil twin attack detection system.

To address the limitations of initializing the system before arriving to the hotspot to disable autoconnection to previous open public Wi-Fi connections, waiting for a disassociated wireless card event to rescan the wireless network, costs of creating a fake remote server to test the validity of a public website certificate, and creating a user interface for a large scale evaluation in the wild, the author offered several recommendations. These recommendations included disabling autoconnect when the user install the system, rescanning the wireless network when the signal level of an AP is below a determined threshold, testing SSL exceptions with a larger set of invalid certificates, and creating an effective graphical user interface. Adoption of these recommendations can provide overall improvements on the client-side mobile evil twin attack detection method and facilitate broader adoption in field studies of a similar nature.

To provide visibility and distribution of the CSMETAD system design, implementation, and evaluation process developed in this study, specifications and procedures are communicated in the form of a solution manual that is available to all academic institutions. Refer to Appendix D.

# Appendix A

## Requirements and limitations mapping based on literature review

| Requirements | Han et al. (2009) | Song et al. (2010) | Han et al. (2011) | Monica & Ribeiro (2011) | Song et al. (2012) | Nikbakhsh et al. (2012) | Kim et al. (2012) | Lanze et al. (2014) | Hsu et al. (2015) | Szongott et al. (2015) | Hossen & Wenyuan (2014) | Nakhila et al. (2015) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. Timing-based or traceroute | Y- | Y- | Y- | N | Y- | Y- | N | N | N | N | Y- | N |
| 2. Assume the attacker uses the legitimate wireless network gateway | Y- | Y- | Y- | Y- | Y- | Y- | Y- | N | Y- | n.a. | Y- | N |
| 3. Assume the attacker uses a different gateway from a legitimate AP | N | N | N | N | N | N | N | Y | N | n.a. | Y | Y |
| 4. Assume attacker performs a mobile attack | N | N | N | N | N | N | N | Y | N | n.a. | Y | Y |
| 5. Require network administrator assistance or privileges | N | N | N | N | N | N | N | N | N | N | N | N |
| 6. System is automated with no intervention from users | N- | N- | N- | Y | N- | n.a. | Y | n.a. | N- | Y | Y | Y |
| 7. Require knowledge of wireless hotspot network infrastructure, AP authorization list and/or user/hosts (trained knowledge) | Y- | Y-<br>(TMM)<br>N<br>(HDT) | Y- | N | Y-<br>(TMM)<br>N<br>(HDT) | N | N | N | N | Y- | N | N |

(continued)

| Requirements | Han et al. (2009) | Song et al. (2010) | Han et al. (2011) | Monica & Ribeiro (2011) | Song et al. (2012) | Nikbakhsh et al. (2012) | Kim et al. (2012) | Lanze et al. (2014) | Hsu et al. (2015) | Szongott et al. (2015) | Hossen & Wenyuan (2014) | Nakhila et al. (2015) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8. Require infrastructure support (hotspot wireless network modification, extra devices/addl. equipment, etc.) | N | Y- | N | Y- | Y- | N | N | N | Y- | Y- | N | N |
| 9. Leverage a public server | N | N | N | N | N | N | N | N | N | N | Y | Y |
| 10. Work with any type of IEEE 802.11 based wireless networks | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 11. Work with Wi-Fi enabled device | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 12. Work with free open (unencrypted) public Wi-Fi networks | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 13. Technology independent (e.g. received signal strength fluctuation, network saturation, network traffic conditions, etc.) | N- | N- | N- | Y | N- | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. |
| 14. Assume that the BSSID of the hotspot APs are unique and use that as a reference to switch between different APs with the same SSID in the hotspot | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | Y- | Y- |

(continued)

| Requirements | Han et al. (2009) | Song et al. (2010) | Han et al. (2011) | Monica & Ribeiro (2011) | Song et al. (2012) | Nikbakhsh et al. (2012) | Kim et al. (2012) | Lanze et al. (2014) | Hsu et al. (2015) | Szongott et al. (2015) | Hossen & Wenyuan (2014) | Nakhila et al. (2015) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15. Assume that the evil twin AP is in the same subnet as the legitimate AP | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | N- | Y |
| 16. Protect users when the attacker blocks access to the public website used to get ISP information | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | N- | N- |
| 17. Protect users when the attacker presents an invalid certificate while retrieving ISP information from public website | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | N- | Y |
| 18. Assume detection of all the APs in the public Wi-Fi network during the initial wireless network scanning | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- |
| 19. Assume that the client is able to associate to all the APs in the public Wi-Fi network | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- | Y- |
| 20. Assume that the user has or has not connected to the target public Wi-Fi network in the past | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- |
| 21. Assume the attacker is connected when the wireless user connects to the public Wi-Fi hotspot | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |

(continued)

| Requirements | Han et al. (2009) | Song et al. (2010) | Han et al. (2011) | Monica & Ribeiro (2011) | Song et al. (2012) | Nikbakhsh et al. (2012) | Kim et al. (2012) | Lanze et al. (2014) | Hsu et al. (2015) | Szongott et al. (2015) | Hossen & Wenyuan (2014) | Nakhila et al. (2015) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 22. Assume the attacker is not connected when the user connects initially to the public Wi-Fi hotspots and protect the wireless user for the duration to the public Wi-Fi network connection | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- |
| 23. Warn the wireless user of an evil twin attack before any traffic is transmitted | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 24. After evil twin detection, the system - connects the user to a legitimate AP | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- |
| 25. Evaluated in lab setting | Y | Y | Y | Y | Y | N | Y | Y | Y | Y | N | Y |
| 26. Evaluated in the field | Y | Y | Y | Y | Y | N | Y | N | Y | N | Y | N |
| 27. Used their own evil twin AP in the evaluation | Y | Y | Y | Y | Y |  | Y | Y | Y | Y | Y | Y |
| 28. Aimed at detecting real evil twin APs (wild) | N- | N- | N- | N- | N- |  | N- |  | N- |  | N- |  |
| 29. Public Wi-Fi hotspots | University | University | University | University | University |  | University<br>Cafes |  | University |  | University<br>Cafes<br>Restaurants<br>Airports |  |
| 30. Large scale evaluation | N | N | N | N | N |  | n.a. |  | N |  | Y |  |

(continued)

| Requirements | Han et al. (2009) | Song et al. (2010) | Han et al. (2011) | Monica & Ribeiro (2011) | Song et al. (2012) | Nikbakhsh et al. (2012) | Kim et al. (2012) | Lanze et al. (2014) | Hsu et al. (2015) | Szongott et al. (2015) | Hossen & Wenyuan (2014) | Nakhila et al. (2015) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 31. Evaluated for performance | Y<br>(Detection Accuracy, Efficiency) | Y<br>(Effectiveness, Efficiency) | Y<br>(Detection Accuracy, Efficiency) | Y<br>(Effectiveness, Efficiency) | Y<br>(Effectiveness, Efficiency) | | Y<br>(Accuracy, True Positive Rate (TPR), False Positive Rate (FPR) | | Y* | | Y<br>(Accuracy, Precision, Recall) | |
| 32. Language | n.a. | n.a. | n.a. | n.a. | n.a. | | n.a. | n.a. | n.a. | Java | n.a. | C |
| 33. Client Platform | laptop | laptop | laptop | laptop | laptop | | smartphone | laptop | laptop | n.a. | smartphone | laptop |
| 34. Client OS | Linux | n.a. | Linux | Linux | n.a. | | Android | n.a. | Windows | n.a. | Android | Linux |

*Note*. Y = included in solution based on literature; N = not included in solution based on literature; n.a. = could not be determined from the literature; - = solution limitation; * = no details provided; blank = not applicable.

# Appendix B

## Algorithm Test Cases and Results

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 1 | 1 | System disables "auto-connections" to open public Wi-Fi networks | • System disabled auto-connections | As Expected | Pass |
| 2 | 1 | System scans public Wi-Fi network | • System displayed list of SSIDs in range | As Expected | Pass |
| 3 | 1 | User selects open public Wi-Fi hotspot SSID | • System created list of APs for selected SSID (signal strength ≥ -75 dBm) | As Expected | Pass |
| 4 | 1 | Number of APs for selected SSID is less than 2 | • System displayed message: "There is insufficient information to detect ETAs";<br>• System ended | As Expected | Pass |
| 5 | 1 | Number of APs for selected SSID is equal to or greater than 2 | • System displayed message: "There is sufficient information to detect ETAs";<br>• System started to iterate across all the APs in the **APs for selected SSID** list | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 6 | 1 | AP iterator is greater than the number of APs for selected SSID | • System ended iteration;<br>• System proceeded to determine trusted global IP address to be used for the duration of the public Wi-Fi connection | As Expected | Pass |
| 7 | 1 | AP iterator is less than the number of APs for selected SSID | • System attempted to associate to AP on **APs for selected SSID** list | As Expected | Pass |
| 8 | 1 | System is not able to associate to an AP | • System updated **APs for selected SSID** list with AP state = "unknown";<br>• System associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |
| 9 | 1 | System is able to associate to AP | • System proceeded to obtain Client DHCP address for the user | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 10 | 1 | System is not able to get Client DHCP address for the user | • System updated **APs for selected SSID** list with AP state = “unknown”;<br>• System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |
| 11 | 1 | System is able to get Client DHCP address for the user | • System proceeded to accept terms of use to access the Internet | As Expected | Pass |
| 12 | 1 | System is not able to accept terms of use to access the Internet | • System updated **APs for selected SSID** list with AP state = “unknown”;<br>• System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |
| 13 | 1 | System is able to accept terms of use to access the Internet | • System proceeded to access secured public website to retrieve global IP address of the AP | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 14 | 1 | System is not able to access secured public website | • System updated **APs for selected SSID** list with AP state = "unknown";<br>• System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |
| 15 | 1 | System is able to access secured public website | • System proceeded to verify that the public website certificate is valid | As Expected | Pass |
| 16 | 1 | Public website certificate is invalid | • System updated **APs for selected SSID** list with AP state = "ETA";<br>• System added AP MAC address to the **learned ETA MAC address** list;<br>• System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 17 | 1 | Public website certificate is valid | • System proceeded to get the global IP address of the AP | As Expected | Pass |
| 18 | 1 | Number of occurrences of a global IP address is less than 2 | • System displayed message: “There is not enough information to categorize APs”;<br>• System updated **APs for selected SSID** list with AP state = “unknown”;<br>• System ended | As Expected | Pass |
| 19 | 1 | Number of occurrences of a global IP address is equal to or greater than 2 | • System displayed message: “There is enough information to categorize APs”;<br>• System set global IP as the trusted global IP address;<br>• System started to categorize APs;<br>• System started to iterate across all the APs in the **APs for selected SSID** list | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 20 | 1 | AP iterator is greater than the number of APs for selected SSID | • System ended iteration;<br>• System moved to Phase 2 detection and protection | As Expected | Pass |
| 21 | 1 | AP iterator is less than the number of APs for selected SSID | • System validated that the global IP address is the same as the trusted global IP address | As Expected | Pass |
| 22 | 1 | Global IP address for an AP is not the same as the trusted global IP address | • System updated **APs for selected SSID** list with AP state = ETA;<br>• System added AP MAC address to the **learned ETA MAC address** list;<br>• System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 23 | 1 | Global IP address for an AP is the same as the trusted global IP address | • System updated **APs for selected SSID** list with AP state = Valid;<br>• System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 1 | 2 | System rescans public Wi-Fi network to rediscover APs with selected SSID | • System displayed list of SSIDs in range;<br>• System created list of APs for selected SSID (signal strength ≥ -75 dBm) (new list) | As Expected | Pass |
| 2 | 2 | Number of APs for selected SSID is less than 1 | • System displayed message: "Your device is out of range for the selected public Wi-Fi hotspot. Please move closer";<br>• Back to step 1 | As Expected | Pass |
| 3 | 2 | Number of APs for selected SSID is equal to or greater than 1 | • System retrieved learned ETA MAC addresses from **Learned ETA MAC addresses** list;<br>• System removed learned ETA MAC addresses from **APs for selected SSID** list | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 4 | 2 | Number of APs for selected SSID is less than 1 | • System displayed message: “You are located on the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi Hotspot”;<br>• Back to step 1 | As Expected | Pass |
| 5 | 2 | Number of APs for selected SSID is equal to or greater than 1 | • System started to iterate across all the APs in the **APs for selected SSID** list | As Expected | Pass |
| 6 | 2 | AP iterator is less than the number of APs for selected SSID | • System rescanned public Wi-Fi network;<br>• Back to step 1 | As Expected | Pass |
| 7 | 2 | AP iterator is greater than the number of APs for selected SSID | • System attempted to associate to the AP with highest signal strength in the **APs for selected SSID** list | As Expected | Pass |
| 8 | 2 | System is not able to associate to the AP | • System associated to the next AP with the highest signal strength in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 9 | 2 | System is able to associate to an AP | • System proceeded to obtain Client DHCP address for the user | As Expected | Pass |
| 10 | 2 | System is not able to get a Client DHCP address for the user | • System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |
| 11 | 2 | System is able to get Client DHCP address for the user | • System proceeded to confirm Internet access | As Expected | Pass |
| 12 | 2 | System is not able to confirm Internet access | • System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |
| 13 | 2 | System is able to confirm Internet access | • System proceeded to access secured public website to retrieve global IP address of the AP | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 14 | 2 | System is not able to access secured public website | • System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |
| 15 | 2 | System is able to access secured public website | • System proceeded to verify that the public website certificate is valid | As Expected | Pass |
| 16 | 2 | Public website certificate is invalid | • System displayed message: "CSMETAD has detected an ETA on the public Wi-Fi hotspot";<br>• System added AP MAC address to the **learned ETA MAC address** list;<br>• System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 17 | 2 | Public website certificate is valid | • System proceeded to get the global IP address of the AP | As Expected | Pass |
| 18 | 2 | Global IP address for an AP is not the same as the trusted global IP address | • System displayed message: “CSMETAD has detected an ETA on the public Wi-Fi hotspot”;<br>• System added AP MAC address to the **learned ETA MAC address** list;<br>• System disassociated from current AP and associated to the next AP in the **APs for selected SSID** list;<br>• Back to step 6/7 | As Expected | Pass |
| 19 | 2 | Global IP address for an AP is the same as the trusted global IP address | • System displayed message: “Wi-Fi connection is safe. You are connected to a legitimate AP”;<br>• System ended iteration;<br>• System waited for a disassociated wireless card event | As Expected | Pass |

(continued)

| Test ID | Phase | Test Step | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 20 | 2 | System receives a disassociated wireless card event | • System rescanned public Wi-Fi network;<br>• Back to step 1 | As Expected | Pass |

# Appendix C

## CSMETAD System Results – Key Requirements

Included herein are the CSMETAD system results for each of the key requirements. This output was generated by Netbeans.

### *R1: It will protect users whether or not they have connected to a free open public Wi-Fi network in the past.*

```
Setting all connections with open security to autoconnect:no
---------------------------------------------------------------
nmcli -f UUID,NAME,TYPE,AUTOCONNECT,AUTOCONNECT-PRIORITY,READONLY,ACTIVE,DEVICE,STATE connection show
UUID                                  NAME                    TYPE             AUTOCONNECT  AUTOCONNECT-PRIORITY  READONLY  ACTIVE  DEVICE  STATE
5664862b-502c-4199-954f-209819f1ffc5  labwifi                 802-11-wireless  yes          0                     no        no      --      --


nmcli --fields connection.id,connection.type,connection.autoconnect,802-11-wireless.ssid,802-11-wireless.mode,802-11-wireless.channel,802-11-
wireless.seen-bssids,802-11-wireless.bssid,802-11-wireless-security.key-mgmt connection show 5664862b-502c-4199-954f-209819f1ffc5

connection.id:                          labwifi
connection.type:                        802-11-wireless
connection.autoconnect:                 yes
802-11-wireless.ssid:                   labwifi
802-11-wireless.mode:                   infrastructure
802-11-wireless.channel:                0
802-11-wireless.bssid:                  --
802-11-wireless.seen-bssids:            58:BC:27:93:05:60

***Found an 802.11 connection with open security. Setting autoconnect: no***
nmcli con mod 5664862b-502c-4199-954f-209819f1ffc5 connection.autoconnect no
```

*R2: It will protect the user when not all the hotspot APs with the desired SSID are detected during initial wireless network scanning.*

```
Start re-scanning the public Wi-Fi network to rediscover APs with selected SSID
---------------------------------------------------------------------------------


Listing SSIDs in range on target OS linux


---------highestRssiApPerEssidBiDemArrList sorted by signal level, and Encryption off at top------------------------------------------
  SSID                                   RSSI              Encryption                 MAC address
labwifi                                   -76                  off                  58:BC:27:93:05:60
Malecon2018                               -36                  on                   B0:7F:B9:74:52:CD
maleconJAL                                -47                  on                   B0:7F:B9:81:28:AB


Listing Aps per Select SSID on target OS linux

  SSID                   RSSI   Encryption     MAC address            Frequency
labwifi                   -76      off      58:BC:27:93:05:60    2.412 GHz (Channel 1)        Signal Level -76 <  threshold -75.
Not Add to apsPerEssidBiDemArrList
labwifi                   -80      off      58:BC:27:12:0C:10    2.462 GHz (Channel 11)       Signal Level -80 <  threshold -75.
Not Add to apsPerEssidBiDemArrList


---------apsPerEssidBiDemArrList sorted by signal level, above threshold--------------------------------------------------------------
Empty


End re-scanning the public Wi-Fi network

Your device is out of range for the selected public Wi-Fi hotspot.  Please move closer.

Start re-scanning the public Wi-Fi network to rediscover APs with selected SSID

Listing SSIDs in range on target OS linux


---------highestRssiApPerEssidBiDemArrList sorted by signal level, and Encryption off at top------------------------------------------
  SSID                                   RSSI              Encryption                 MAC address
labwifi                                   -41                  off                  58:BC:27:93:05:60
Malecon2018                               -37                  on                   B0:7F:B9:74:52:CD
maleconJAL                                -46                  on                   B0:7F:B9:81:28:AB
```

```
Listing Aps per Select ESSID on target OS linux

  SSID                   RSSI   Encryption     MAC address            Frequency
labwifi                   -41      off      58:BC:27:93:05:60    2.412 GHz (Channel 1)        Signal Level -41 >= threshold -75.
Add to apsPerEssidBiDemArrList
labwifi                   -57      off      58:BC:27:12:0C:10    2.462 GHz (Channel 11)       Signal Level -57 >= threshold -75.
Add to apsPerEssidBiDemArrList


---------apsPerEssidBiDemArrList sorted by signal level, above threshold--------------------------------------------------------------
  SSID                RSSI    Encryption      MAC address             Frequency
labwifi                -41      off        58:BC:27:93:05:60     2.412 GHz (Channel 1)
labwifi                -57      off        58:BC:27:12:0C:10     2.462 GHz (Channel 11)


End re-scanning the public Wi-Fi network
```

*R3: It will protect the user when the client is not able to associate to all the APs in the public Wi-Fi network.*

```
Current AP MAC Address:58:BC:27:93:05:60    signal level:-37

Associate client to current AP
--------------------------------
iwconfig wlp3s0 mode managed essid labwifi ap 58:BC:27:93:05:60

Checking association status
iwconfig wlp3s0
wlp3s0    IEEE 802.11  ESSID:"labwifi"
          Mode:Managed  Access Point: Not-Associated   Tx-Power=15 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off


inspect Association (1/5).Found value: Not-Associated

sleep 1 second

Checking association status
iwconfig wlp3s0
wlp3s0    IEEE 802.11  ESSID:"labwifi"
          Mode:Managed  Access Point: Not-Associated   Tx-Power=15 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off


inspect Association (2/5).Found value: Not-Associated

sleep 1 second

Checking association status
iwconfig wlp3s0
wlp3s0    IEEE 802.11  ESSID:"labwifi"
          Mode:Managed  Access Point: Not-Associated   Tx-Power=15 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off


inspect Association (3/5).Found value: Not-Associated

sleep 1 second
```

```
Checking association status
iwconfig wlp3s0
wlp3s0    IEEE 802.11  ESSID:"labwifi"
          Mode:Managed  Access Point: Not-Associated   Tx-Power=15 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off


inspect Association (4/5).Found value: Not-Associated

sleep 1 second

Checking association status
iwconfig wlp3s0
wlp3s0    IEEE 802.11  ESSID:"labwifi"
          Mode:Managed  Access Point: Not-Associated   Tx-Power=15 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off


inspect Association (5/5).Found value: Not-Associated

sleep 1 second


Client was not able to associate to AP


CSMETAD will try to connect to another AP if available
```

*R4 & R7 & R8: It will protect users when the attacker sets up the mobile evil twin AP with the same SSID, BSSID and subnet of a legitimate AP. After detection, it connects the user to a legitimate AP. Lastly, it protects the user for the duration of the public Wi-Fi connection, discovering and reporting on new mobile evil twin access points.*

```
Start re-scanning the public Wi-Fi network to rediscover APs with selected SSID
---------------------------------------------------------------------------------


Listing SSIDs in range on target OS linux


---------highestRssiApPerEssidBiDemArrList sorted by signal level, and Encryption off at top------------------------------------------
  SSID                                   RSSI              Encryption                MAC address
labwifi                                   -42                  off                 58:BC:27:93:05:60
Malecon2018                               -37                  on                  B0:7F:B9:74:52:CD
maleconJAL                                -46                  on                  B0:7F:B9:81:28:AB


Listing APs for selected SSID on target OS linux

  SSID                   RSSI   Encryption     MAC address            Frequency
labwifi                   -42      off      58:BC:27:93:05:60    2.412 GHz (Channel 1)        Signal Level -42 >= threshold -75.
Add to apsPerEssidBiDemArrList
labwifi                   -78      off      58:BC:27:12:0C:10    2.462 GHz (Channel 11)       Signal Level -78 <  threshold -75.
Not Add to apsPerEssidBiDemArrList


---------apsPerEssidBiDemArrList sorted by signal level, above threshold--------------------------------------------------------------
  SSID                RSSI    Encryption      MAC address             Frequency
labwifi                -42      off        58:BC:27:93:05:60     2.412 GHz (Channel 1)


End re-scanning the public Wi-Fi network

Showing learned ETA MAC address
Empty

Remove learned ETA MAC addresses from re-scanned APs for selected SSID list
```

```
---------AsPerEssidBiDemArrList AP MAC addresses, sorted by signal level, above threshold---------------------------------------------
  SSID                RSSI   Encryption     MAC address              Frequency
labwifi                -42     off        58:BC:27:93:05:60     2.412 GHz (Channel 1)


Now iterating across APs
--------------------------


Current AP MAC Address:58:BC:27:93:05:60    signal level:-42


Associate client to current AP
--------------------------------
iwconfig wlp3s0 mode managed essid labwifi ap 58:BC:27:93:05:60

Checking association status
iwconfig wlp3s0
wlp3s0    IEEE 802.11  ESSID:"labwifi"
          Mode:Managed  Frequency:2.412 GHz  Access Point: 58:BC:27:93:05:60
          Bit Rate=1 Mb/s   Tx-Power=15 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=68/70  Signal level=-42 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:1  Invalid misc:0   Missed beacon:0


Client has associated to AP


Get client DHCP address
-------------------------
dhclient -timeout 20 wlp3s0

Checking if client has a valid IP address
-------------------------------------------
ifconfig wlp3s0
wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.37  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::8e70:5aff:fe82:9264  prefixlen 64  scopeid 0x20<link>
        ether 8c:70:5a:82:92:64  txqueuelen 1000  (Ethernet)
        RX packets 213  bytes 162995 (159.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 232  bytes 31821 (31.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
Found IP address: 192.168.43.37

cat /etc/resolv.conf
nameserver 8.8.8.8


Confirm Internet access
-------------------------
Attempting to access URL to check Internet connection
URL response code: 200
First attempt to detect if client is behind a captive portal
HTML content omitted
A captive portal was not detected
Internet access confirmed


Access secured public website and verify if public website certificate is valid
---------------------------------------------------------------------------------
Response Code : 200
Cipher Suite : TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256


Cert Type : X.509
Cert Hash Code : 1749875764
Cert Public Key Algorithm : RSA
Cert Public Key Format : X.509


Cert Type : X.509
Cert Hash Code : -2059616493
Cert Public Key Algorithm : RSA
Cert Public Key Format : X.509


Cert Type : X.509
Cert Hash Code : 1215155824
Cert Public Key Algorithm : RSA
Cert Public Key Format : X.509
```

```
Now printing secured public website object access, certificate status, global IP
----------------------------------------------------------------------------------
Access: true
Certificate status: valid
Global IP: 174.194.14.15
----------------------------------------------------------------------------------
CSMETAD has detected an ETA on the public Wi-Fi hotspot

CSMETAD added ETA MAC address to learned ETA MAC address list

CSMETAD will try to connect to another AP if available

CSMETAD finished iterating across the list of APs

CSMETAD was not able to validate that the AP global IP was the same as the trusted global IP

CSMETAD will scan the public Wi-Fi network again


Start re-scanning the public Wi-Fi network to rediscover APs with selected SSID
---------------------------------------------------------------------------------


Listing SSIDs in range on target OS linux


---------highestRssiApPerEssidBiDemArrList sorted by signal level and encryption off at top-------------------------------------------
  SSID                                    RSSI              Encryption                 MAC address
labwifi                                   -9                   off                  58:BC:27:93:05:60
Malecon2018                               -37                  on                   B0:7F:B9:74:52:CD
maleconJAL                                -46                  on                   B0:7F:B9:81:28:AB


Listing APs for selected SSID on target OS linux


  SSID                    RSSI  Encryption     MAC address             Frequency
labwifi                   -9      off      58:BC:27:93:05:60    2.412 GHz (Channel 1)        Signal Level -9  >= threshold -75.
Add to apsPerEssidBiDemArrList
labwifi                   -78     off      58:BC:27:12:0C:10    2.462 GHz (Channel 11)       Signal Level -78 <  threshold -75.
Not Add to apsPerEssidBiDemArrList
```

```
---------apsPerEssidBiDemArrList sorted by signal level, above threshold--------------------------------------------------------------
  SSID                RSSI    Encryption      MAC address              Frequency
labwifi                -9       off        58:BC:27:93:05:60     2.412 GHz (Channel 1)


End re-scanning the public Wi-Fi network

Showing learned ETA MAC address
58:BC:27:93:05:60

Remove learned ETA MAC addresses from re-scanned APs for selected SSID list

You are located on the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi hotspot.

Scanning the public Wi-Fi network again


Start re-scanning the public Wi-Fi network to rediscover APs with selected SSID
---------------------------------------------------------------------------------


Listing SSIDs in range on target OS linux


---------highestRssiApPerEssidBiDemArrList sorted by signal level and encryption off at top-------------------------------------------
  SSID                                    RSSI              Encryption                 MAC address
labwifi                                   -9                   off                  58:BC:27:93:05:60
Malecon2018                               -37                  on                   B0:7F:B9:74:52:CD
maleconJAL                                -46                  on                   B0:7F:B9:81:28:AB



Listing APs for selected SSID on target OS linux

  SSID                   RSSI   Encryption     MAC address             Frequency
labwifi                   -9      off      58:BC:27:93:05:60    2.412 GHz (Channel 1)        Signal Level -9  >=  threshold -75.
Add to apsPerEssidBiDemArrList
labwifi                   -53     off      58:BC:27:12:0C:10    2.462 GHz (Channel 11)       Signal Level -53 >=  threshold -75.
Add to apsPerEssidBiDemArrList


---------apsPerEssidBiDemArrList sorted by signal level, above threshold--------------------------------------------------------------
  SSID                RSSI    Encryption      MAC address              Frequency
labwifi                -9        off        58:BC:27:93:05:60     2.412 GHz (Channel 1)
labwifi                -53       off        58:BC:27:12:0C:10     2.462 GHz (Channel 11)
```

```
End re-scanning the public Wi-Fi network

Showing learned ETA MAC address
58:BC:27:93:05:60

Remove learned ETA MAC addresses from re-scanned APs for selected SSID list

---------AsPerEssidBiDemArrList AP MAC addresses, sorted by signal level, above threshold---------------------------------------------
  SSID                RSSI    Encryption      MAC address             Frequency
labwifi                -53      off        58:BC:27:12:0C:10     2.462 GHz (Channel 11)


Now iterating across APs
--------------------------
Current AP MAC Address:58:BC:27:12:0C:10    signal level:-53


Associate client to current AP
--------------------------------
iwconfig wlp3s0 mode managed essid labwifi ap 58:BC:27:12:0C:10

Checking association status
iwconfig wlp3s0
wlp3s0    IEEE 802.11  ESSID:"labwifi"
          Mode:Managed  Frequency:2.462 GHz  Access Point: 58:BC:27:12:0C:10
          Bit Rate=1 Mb/s   Tx-Power=15 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=55/70  Signal level=-55 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0


Client has associated to AP


Get client DHCP address
-------------------------
dhclient -timeout 20 wlp3s0

Checking if client has a valid IP address
-------------------------------------------
ifconfig wlp3s0
wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.37  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::8e70:5aff:fe82:9264  prefixlen 64  scopeid 0x20<link>
        ether 8c:70:5a:82:92:64  txqueuelen 1000  (Ethernet)
        RX packets 266  bytes 200613 (195.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 298  bytes 40803 (39.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

Found IP address: 192.168.43.37

cat /etc/resolv.conf
nameserver 209.244.0.3
nameserver 209.244.0.4

Confirm Internet access
-------------------------
Attempting to access URL to check Internet connection
URL response code: 200
First attempt to detect if client is behind a captive portal
HTML content omitted
A captive portal was not detected
Internet access confirmed


Access secured public website and verify if public website certificate is valid
---------------------------------------------------------------------------------
Response Code : 200
Cipher Suite : TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256


Cert Type : X.509
Cert Hash Code : 1749875764
Cert Public Key Algorithm : RSA
Cert Public Key Format : X.509


Cert Type : X.509
Cert Hash Code : -2059616493
Cert Public Key Algorithm : RSA
Cert Public Key Format : X.509


Cert Type : X.509
Cert Hash Code : 1215155824
Cert Public Key Algorithm : RSA
Cert Public Key Format : X.509
```

```
Now printing secured public website object access, certificate status, global IP
----------------------------------------------------------------------------------
Access: true
Certificate status: valid
Global IP: 173.95.190.140
----------------------------------------------------------------------------------
Wi-Fi connection is safe.  You are connected to a legitimate AP.

running iwevent. waiting for event "Not-Associated"

Waiting for Wireless Events from interfaces...
```

*R5:  It protects the user when the attacker blocks access to the public website used to get ISP information.*

```
Access secured public website and verify if public website certificate is valid
----------------------------------------------------------------------------------
Some other exception thrown:
java.net.ConnectException: Connection refused (Connection refused)
	at java.net.PlainSocketImpl.socketConnect(Native Method)
	at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
	at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
	at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
	at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
	at java.net.Socket.connect(Socket.java:589)
	at sun.security.ssl.SSLSocketImpl.connect(SSLSocketImpl.java:668)
	at sun.security.ssl.BaseSSLSocketImpl.connect(BaseSSLSocketImpl.java:173)
	at sun.net.NetworkClient.doConnect(NetworkClient.java:180)
	at sun.net.www.http.HttpClient.openServer(HttpClient.java:463)
	at sun.net.www.http.HttpClient.openServer(HttpClient.java:558)
	at sun.net.www.protocol.https.HttpsClient.<init>(HttpsClient.java:264)
	at sun.net.www.protocol.https.HttpsClient.New(HttpsClient.java:367)
	at sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.getNewHttpClient(AbstractDelegateHttpsURLConnection.java:191)
	at sun.net.www.protocol.http.HttpURLConnection.plainConnect0(HttpURLConnection.java:1138)
	at sun.net.www.protocol.http.HttpURLConnection.plainConnect(HttpURLConnection.java:1032)
	at sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.connect(AbstractDelegateHttpsURLConnection.java:177)
	at sun.net.www.protocol.http.HttpURLConnection.getInputStream0(HttpURLConnection.java:1546)
	at sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1474)
	at java.net.HttpURLConnection.getResponseCode(HttpURLConnection.java:480)
	at sun.net.www.protocol.https.HttpsURLConnectionImpl.getResponseCode(HttpsURLConnectionImpl.java:338)
	at clientsidemobileeviltwinattackdetection.SecurePublicIpSite.detectGlobalIp(SecurePublicIpSite.java:51)
	at 
clientsidemobileeviltwinattackdetection.ClientSideMobileEvilTwinAttackDetection.main(ClientSideMobileEvilTwinAttackDetection.java:625)


Now printing secured public website object access, certificate status, global IP
-----------------------------------------------------------------------------------
Access: false
Certificate status: not detected
Global IP: not detected
-----------------------------------------------------------------------------------

CSMETAD cannot access secured public website

CSMETAD will try to connect to another AP if available
```

*R6: It protects the user when the attacker presents an invalid certificate while retrieving ISP information from a public website.*

```
Access secured public website and verify if public website certificate is valid
----------------------------------------------------------------------------------
SSL exception thrown:
javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
        at sun.security.ssl.Alerts.getSSLException(Alerts.java:192)
        at sun.security.ssl.SSLSocketImpl.fatal(SSLSocketImpl.java:1949)
        at sun.security.ssl.Handshaker.fatalSE(Handshaker.java:302)
        at sun.security.ssl.Handshaker.fatalSE(Handshaker.java:296)
        at sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1514)
        at sun.security.ssl.ClientHandshaker.processMessage(ClientHandshaker.java:216)
        at sun.security.ssl.Handshaker.processLoop(Handshaker.java:1026)
        at sun.security.ssl.Handshaker.process_record(Handshaker.java:961)
        at sun.security.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:1062)
        at sun.security.ssl.SSLSocketImpl.performInitialHandshake(SSLSocketImpl.java:1375)
        at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:1403)
        at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:1387)
        at sun.net.www.protocol.https.HttpsClient.afterConnect(HttpsClient.java:559)
        at sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.connect(AbstractDelegateHttpsURLConnection.java:185)
        at sun.net.www.protocol.http.HttpURLConnection.getInputStream0(HttpURLConnection.java:1546)
        at sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1474)
        at java.net.HttpURLConnection.getResponseCode(HttpURLConnection.java:480)
        at sun.net.www.protocol.https.HttpsURLConnectionImpl.getResponseCode(HttpsURLConnectionImpl.java:338)
        at clientsidemobileeviltwinattackdetection.SecurePublicIpSite.detectGlobalIp(SecurePublicIpSite.java:51)
        at
clientsidemobileeviltwinattackdetection.ClientSideMobileEvilTwinAttackDetection.main(ClientSideMobileEvilTwinAttackDetection.java:325)
Caused by: sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
        at sun.security.validator.PKIXValidator.doBuild(PKIXValidator.java:387)
        at sun.security.validator.PKIXValidator.engineValidate(PKIXValidator.java:292)
        at sun.security.validator.Validator.validate(Validator.java:260)
        at sun.security.ssl.X509TrustManagerImpl.validate(X509TrustManagerImpl.java:324)
        at sun.security.ssl.X509TrustManagerImpl.checkTrusted(X509TrustManagerImpl.java:229)
        at sun.security.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustManagerImpl.java:124)
        at sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1496)
        ... 15 more
Caused by: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
        at sun.security.provider.certpath.SunCertPathBuilder.build(SunCertPathBuilder.java:141)
        at sun.security.provider.certpath.SunCertPathBuilder.engineBuild(SunCertPathBuilder.java:126)
        at java.security.cert.CertPathBuilder.build(CertPathBuilder.java:280)
        at sun.security.validator.PKIXValidator.doBuild(PKIXValidator.java:382)
        ... 21 more
```

```
Now printing secured public website object access, certificate status, global IP
----------------------------------------------------------------------------------
Access: true
Certificate status: invalid
global IP: not detected
----------------------------------------------------------------------------------

CSMETAD has detected an ETA on the public Wi-Fi hotspot

CSMETAD added ETA MAC address to learned ETA MAC address list

CSMETAD will try to connect to another AP if available
```

# Appendix D

# CSMETAD System Solution Manual

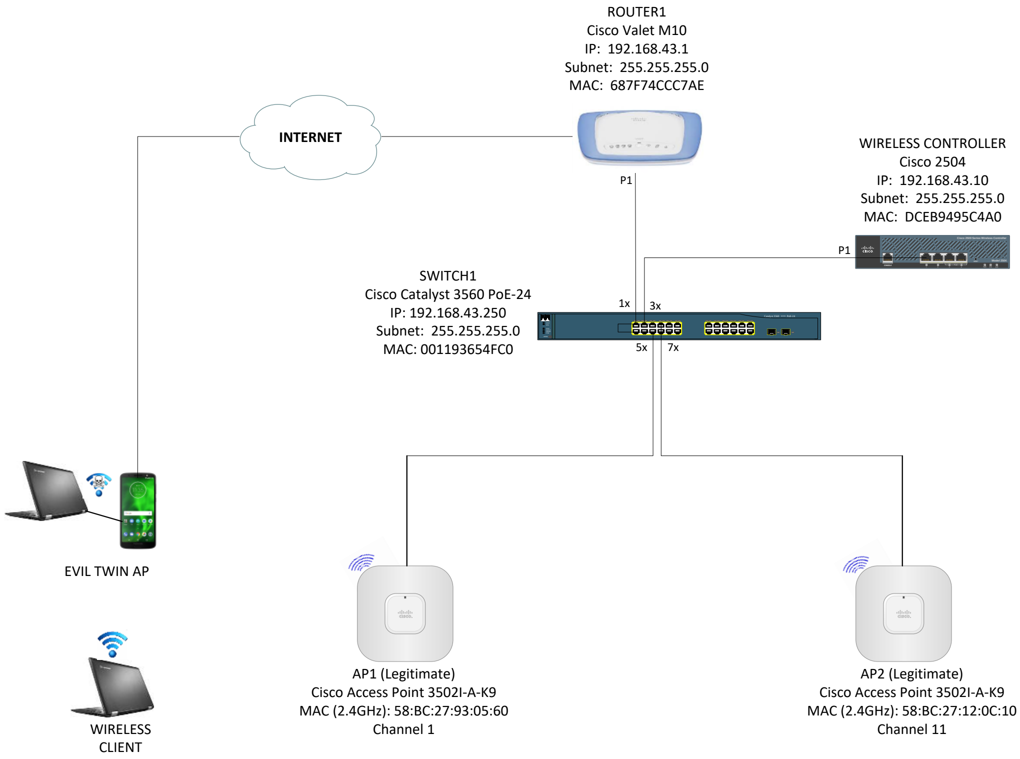
## 1. Physical network connectivity design

The certified equipment included in Hossen & Wenyuan's (2014) study was replaced for the CSMETAD system to expand and provide protection to traveling users that utilize a different mobile platform and operating system in free open public Wi-Fi hotspots. The following is a list of replacement hardware and software included in the design of the CSMETAD system that is central to this dissertation report:

1. *Client Platform:* Hossen & Wenyuan's study's artifact was built for smartphone platforms. CSMETAD was built for laptop platforms. The client laptop platform for this study is a Lenovo Thinkpad laptop.
2. *Client Operating System (OS):* Hossen & Wenyuan's study's artifact was built for Android operating system. CSMETAD was built for Linux operating system. The Linux OS version for this study is 7.3.1611.
3. *Client Programming Language:* Hossen & Wenyuan's study's artifact's programming language was not provided in their study. CSMETAD was built using Java programming language. The Java SE Development Kit is version 1.8.0_131 (64 bits) and the NetBeans Integrated Development Environment is version 8.1.
4. *Mobile Evil Twin AP*: Hossen & Wenyuan (2014) performed the evaluation using a smartphone with mobile AP functionality as the evil twin AP (Nexus 4 Android smartphone with 3G data subscription and Android mobile hotspot and tethering). CSMETAD was evaluated using a laptop and smartphone with mobile AP functionality as the evil twin AP (Lenovo Thinkpad laptop, Kali Linux (Aircrack-ng)

and Hostapd, Motorola Moto e$^{5}$smartphone with 4G data subscription and Android mobile hotspot and tethering).

## 2. Logical prototype topology design diagram



## 3. Artifact construction specifications

### Wireless Client

The hardware and software specifications for the *wireless client* are described as follows:

*Hardware*

The Lenovo Thinkpad Laptop specifications can be retrieved from https://support.lenovo.com/mn/en/solutions/pd027202

*Software*

The Linux OS 7.3.1611 specifications designed for inclusion into the CSMETAD system can be retrieved from https://wiki.centos.org/Manuals/ReleaseNotes/CentOS7

The Java SE Development Kit 1.8.0_131 (64 bits) programming language specifications designed for inclusion into the CSMETAD system can be retrieved from https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf

The NetBeans Integrated Development Environment 8.1 specifications designed for inclusion into the CSMETAD system can be retrieved from https://netbeans.org/community/releases/81/relnotes.html

The Wireshark Packet Analyzer 2.0.0 software was installed in the client specifically for network packet analysis purposes.  The specifications can be retrieved from https://www.wireshark.org/docs/relnotes/wireshark-2.0.0.html

## Mobile Evil Twin AP

The hardware and software specifications for the *mobile evil twin AP* are described as follows:

### Hardware

The Motorola Moto e5 smartphone specifications can be retrieved from https://www.motorola.com/us/products/moto-e-plus-gen-5

The Lenovo Thinkpad Laptop specifications can be retrieved from https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-x/Thinkpad-X1-Carbon-4th-Gen/p/22TP2TXX14G

### Software

The Android Mobile Hotspot and Tethering specifications can be retrieved from https://www.verizonwireless.com/support/knowledge-base-217411/

The Kali Linux 4.14.0 (Aircrack-ng) specifications can be retrieved from https://www.kali.org/news/kali-linux-2018-1-release/

The Hostapd v2.7 specifications can be retrieved from http://w1.fi/hostapd/

## Lab Network

The hardware and software specifications for the *lab network* are described as follows:

The Cisco M10 router specifications documented in an installation guide (Cisco Systems, 2010) can be retrieved from http://downloads.linksys.com/downloads/userguide/1224655305646/Valet_Valet_Plus_M10_M20_UG_US_V10_D-WEB_3425-014530.pdf

The Cisco 3560 switch specifications documented in a spec sheet (Cisco Systems, 2009) can be retrieved from http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3560-series-switches/product_data_sheet09186a00801f3d7d.pdf

The Cisco 2504 wireless controller specifications documented in a spec sheet (Cisco Systems, 2016) can be retrieved from http://www.cisco.com/c/en/us/products/collateral/wireless/2500-series-wireless-controllers/data_sheet_c78-645111.pdf

The Cisco 3502I wireless access point specifications documented in a spec sheet (Cisco Systems, 2012) can be retrieved from http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1250-series/data_sheet_c78-594630.pdf

## 4. Minimum hardware and software requirements

The components and costs to build the CSMETAD system and lab as defined in this dissertation report are listed in below table.

*Evil Twin Detection Lab Environment Components and Costs*

| Component | Quantity | Estimated Cost |
|---|---|---|
| Hardware | | |
| Cisco 3560 Switch | 1 | $150 |
| Cisco Router M10 | 1 | $200 |
| Cisco Wireless Controller 2504 | 1 | $490 |
| Cisco Access Point 3502I-A-K9 (AP1) (legitimate AP) | 1 | $80 |
| Cisco Access Point 3502I-A-K9 (AP2) (legitimate AP) | 1 | $80 |
| Lenovo Laptop | 2 | $2,500 |
| Motorola Moto e$^5$ Android smartphone | 1 | $150 |
| USB wireless adapter | 1 | $40 |
| Ethernet cables | - | $30 |
| Software | | |
| Wireshark Packet Analyzer 2.0.0 | 1 | Free |
| Java SE Development Kit 1.8.0_131 | 1 | Free |
| NetBeans IDE 8.1 | 1 | Free |
| Linux Centos 7.3.1611 | 1 | Free |
| Kali Linux 4.14.0 (Aircrack-ng) | 1 | Free |
| Hostapd v2.7 | 1 | Free |
| Android Mobile Hotspot &Tethering | 1 | Free |
| Switch IOS | 1 | Included |
| Router IOS | 1 | Included |
| Controller IOS | 1 | Included |
| APs IOS | 3 | Included |
| Total | | $3,720 |

## 5. Step-by-step artifact construction procedures

Construction of the CSMETAD system was based primarily on Hevner's principles 1 and 5 through the creation of a viable artifact that relies on the application of rigorous construction methods.

*Lab Environment - Steps:*

1. Unpacking and assembling the equipment.
2. The Linux Centos 7.3.1611 OS was installed and configured in the Lenovo Thinkpad laptop (client) in accordance with the installation guide (Centos, 2016) retrieved from https://wiki.centos.org/Manuals/ReleaseNotes/CentOS7
3. The Java SE Development Kit 1.8.0_131 (64 bits) software was installed and configured in the Lenovo Thinkpad laptop (client) in accordance with the installation guide (Oracle, 2016) retrieved from http://docs.oracle.com/javase/8/docs/technotes/guides/install/index.html
4. The NetBeans Integrated Development Environment 8.1 software was installed and configured in the Lenovo Thinkpad laptop (client) in accordance with the installation guide (Netbeans, 2015) retrieved from https://netbeans.org/community/releases/81/install.html
5. The Wireshark Packet Analyzer 2.0.0 software was installed and configured in the Lenovo Thinkpad laptop (client) in accordance with the installation guide (Wireshark, 2014) retrieved from https://www.wireshark.org/docs/wsug_html/
6. The Kali Linux 4.14.0 (Aircrack-ng) was installed and configured in the Lenovo Thinkpad laptop (ETA) in accordance with the installation guide (Kali, 2018) retrieved from https://docs.kali.org/category/installation

7. The Hostapd v2.7 was installed and configured in the Lenovo Thinkpad laptop (ETA) in accordance with the installation guide (Hostapd, 2013) retrieved from https://w1.fi/hostapd/
8. The Motorola smartphone was configured with tethering in accordance with the instructions (Motorola, 2018) retrieved from https://www.verizonwireless.com/support/knowledge-base-217411/
9. The Cisco M10 router was installed and configured in accordance with the installation guide (Cisco Systems, 2010) retrieved from http://downloads.linksys.com/downloads/userguide/1224655305646/Valet_Valet_Plus_M10_M20_UG_US_V10_D-WEB_3425-014530.pdf
10. The Cisco 3560 switch was installed and configured in accordance with the installation guide (Cisco Systems, 2010) retrieved from http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3560/hardware/installation/guide/3560hig.pdf
11. The Cisco 2504 wireless controller was installed and configured in accordance with the installation guide (Cisco Systems, 2017) retrieved from https://www.cisco.com/c/en/us/support/docs/wireless/2500-series-wireless-controllers/113034-2500-deploy-guide-00.html
12. The Cisco 3502I access points was installed and configured in accordance with the installation guide (Cisco Systems, 2014) retrieved from http://www.cisco.com/c/en/us/td/docs/wireless/access_point/3500/quick/guide/ap3500getstart.pdf

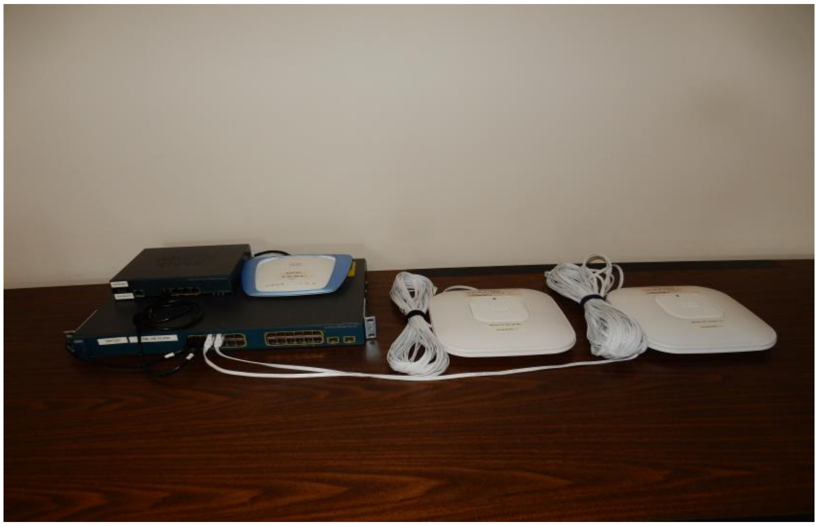


*Figure 9.* Lab network.

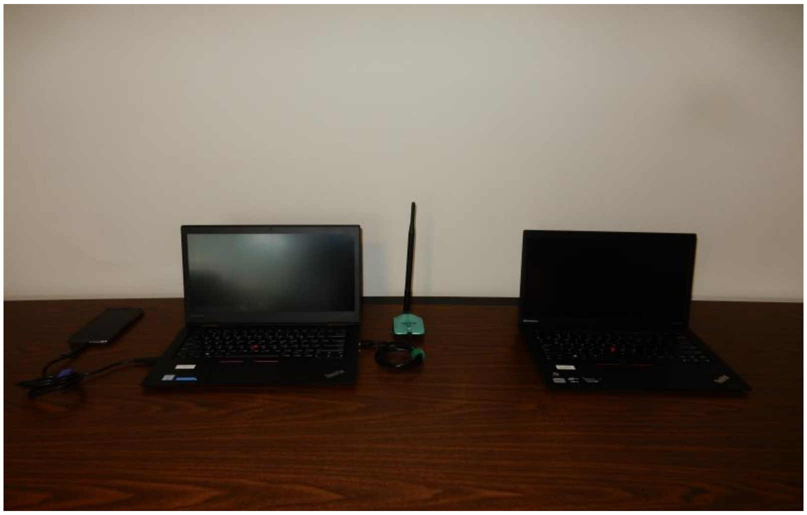


*Figure 10.* Lab wireless client and mobile evil twin AP.

*Client-side Mobile Evil Twin AP Detection (CSMETAD) Algorithm - Steps:*

*Phase 1: Data Collection*

**Basic Flow:**

1. System is initialized by the user before using the free open public Wi-Fi network.
2. System detects operating system.
3. System detects wireless network card.
4. System disables “auto-connections” to all public Wi-Fi networks.
5. System scans the public Wi-Fi network to discover available SSIDs (encrypted and unencrypted).
6. System creates list of **SSIDs in range**.
7. System presents SSIDs in range to the user.
8. User selects unencrypted public Wi-Fi hotspot SSID.
9. System creates list of **APs for selected SSID** with signal level equal to or greater than -75dBm.
10. System validates that the **number of APs for selected SSID** is equal to or greater than 2. **(Alternative Flow “a”)**
11. IF **number of APs for selected SSID** is equal to or greater than 2 THEN System displays message: “There is sufficient information to start detecting ETAs.”
12. System stops the network manager.
13. System activates wireless network card.
14. System starts iterating across all the APs in the **APs for selected SSID** list.
15. System associates to the AP in the **APs for selected SSID** list. **(Alternative Flow “b”)**

16. IF System is able to associate to the AP THEN System gets a Client DHCP address for the user. **(Alternative Flow “c”)**
17. IF System is able to get a Client DHCP address for the user THEN System accepts terms of use to access the Internet. **(Alternative Flow “d”)**
18. IF System is able to accept terms of use to access the Internet THEN System connects to secured public website to retrieve global IP address of the AP. **(Alternative Flow “e”)**
19. IF System is able to connect to the secured public website THEN System verifies that the public website certificate is valid. **(Alternative Flow “f”)**
20. IF the public website certificate is valid only THEN System is able to get the global IP address of the AP.
21. IF System ends iterating across APs in **APs for selected SSID** list THEN System validates that the **number of occurrences of a global IP address** is equal to or greater than 2. **(Alternative Flow “g”)**
22. IF the **number of occurrences of a global IP address** is equal to or greater than 2 THEN System has enough information to start categorizing APs and sets global IP as the trusted global IP address to be used for the duration of the public Wi-Fi connection.
23. System starts categorizing APs.
24. System starts iterating across all the APs in the **APs for selected SSID** list.
25. System validates that the global IP address for an AP is the same as the trusted global IP address. **(Alternative Flow “h”)**

26. IF the global IP address for an AP is the same as the trusted global IP address THEN System categorizes the AP as “valid”.  System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
27. IF System ends iterating across APs in **APs for selected SSID** list THEN System moves to Phase 2 detection and protection.

**Alternative Flows:**

*a) number of APs for selected SSID is less than 2*

On step 10 of the Basic Flow:

1. IF the **number of APs for selected SSID** is less than 2 THEN
2. System displays message: “There is insufficient information to detect ETAs.”
3. System ends.

*b) system is not able to associate to an AP*

On step 15 of the Basic Flow:

1. IF System is not able to associate to an AP THEN
2. System updates the **APs for selected SSID** list with the following results:
    a. Association status = false
    b. Client DHCP address = not detected
    c. Internet access = not detected
    d. Secured public website access = not detected
    e. Certificate status = not detected
    f. Global IP address = not detected
    g. AP state = unknown
3. System associates to the next AP in the **APs for selected SSID** list.

4. Flow of events returns to step 15 of the Basic Flow.
5. System ends.

*c) system is not able to get a Client DHCP address for the user*

On step 16 of the Basic Flow:

1. IF System is not able to get a Client DHCP address for the user THEN
2. System updates the **APs for selected SSID** list with the following results:
    a. Client DHCP address = not detected
    b. Internet access = not detected
    c. Secured public website access = not detected
    d. Certification status = not detected
    e. Global IP address = not detected
    f. AP state = unknown
3. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
4. Flow of events returns to step 15 of the Basic Flow.

*d) system is not able to accept terms of use to access the Internet*

On step 17 of the Basic Flow:

1. IF System is not able to accept terms of use to access the Internet THEN
2. System updates the **APs for selected SSID** list with the following results:
    a. Internet access = false
    b. Secured public website access = not detected
    c. Certification status = not detected
    d. Global IP address = not detected

   e. AP state = unknown

3. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
4. Flow of events returns to step 15 of the Basic Flow.

*e) system is not able to access secured public website*

On step 18 of the Basic Flow:

1. IF System is not able to access secured public website THEN
2. System updates the **APs for selected SSID** list with the following results:
   a. Secured public website access = false
   b. Certification status = not detected
   c. Global IP address = not detected
   d. AP state = unknown
3. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
4. Flow of events returns to step 15 of the Basic Flow.

*f) invalid certificate*

On step 19 of the Basic Flow:

1. IF System receives an invalid certificate message THEN
2. System updates the **APs for selected SSID** list with the following results:
   a. Certification status = invalid
   b. Global IP address = not detected
   c. AP state = ETA
3. System adds AP MAC address to the **learned ETA MAC address** list.

4. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
5. Flow of events returns to step 15 of the Basic Flow.

*g) number of occurrences of a global IP address is less than 2*

On step 21 of the Basic Flow:

1. IF System determines that number of occurrences of a global IP address is less than 2 THEN
2. System displays message: “There is not enough information to categorize APs”
3. System updates the **APs for selected SSID** list with the following result:
    a. AP state = unknown
4. System ends.

*h) global IP address for an AP is not the same as the trusted global IP address*

On step 25 of the Basic Flow:

1. IF System determines that the global IP address for an AP is not the same as the trusted global IP address THEN
2. System categorizes the AP as “ETA”.
3. System updates the **APs for selected SSID** list with the following result:
    a. AP state = ETA
4. System adds the AP MAC addresses to the **learned ETA MAC address** list.
5. System disassociates from current AP and associates to the next AP on the **APs for selected SSID** list.
6. Flow of events returns to step 25 of the Basic Flow.

**Input and Output details:**

1. **SSIDs in range** list = list of AP MAC address, signal level/RSSI, encryption status, frequency, and channel. This list contains APs with encryption on and off.
2. **APs for selected SSID** list = list of AP MAC address, signal level/RSSI, encryption status, frequency, channel, Client DHCP address, Internet access, secured public website access, certification status, global IP address, and AP state. This list contains only the APs with encryption off and signal/RSSI level equal to or greater than -75dBm.
3. **Learned ETA MAC address** list = list of ETA MAC addresses.

**Rule details:**

1. **Number of APs for selected SSID** = the number of APs for selected SSID must be equal to or greater than 2.
2. **Number of occurrences of a global IP address =** the number of occurrences of a global IP address must be equal to or greater than 2.

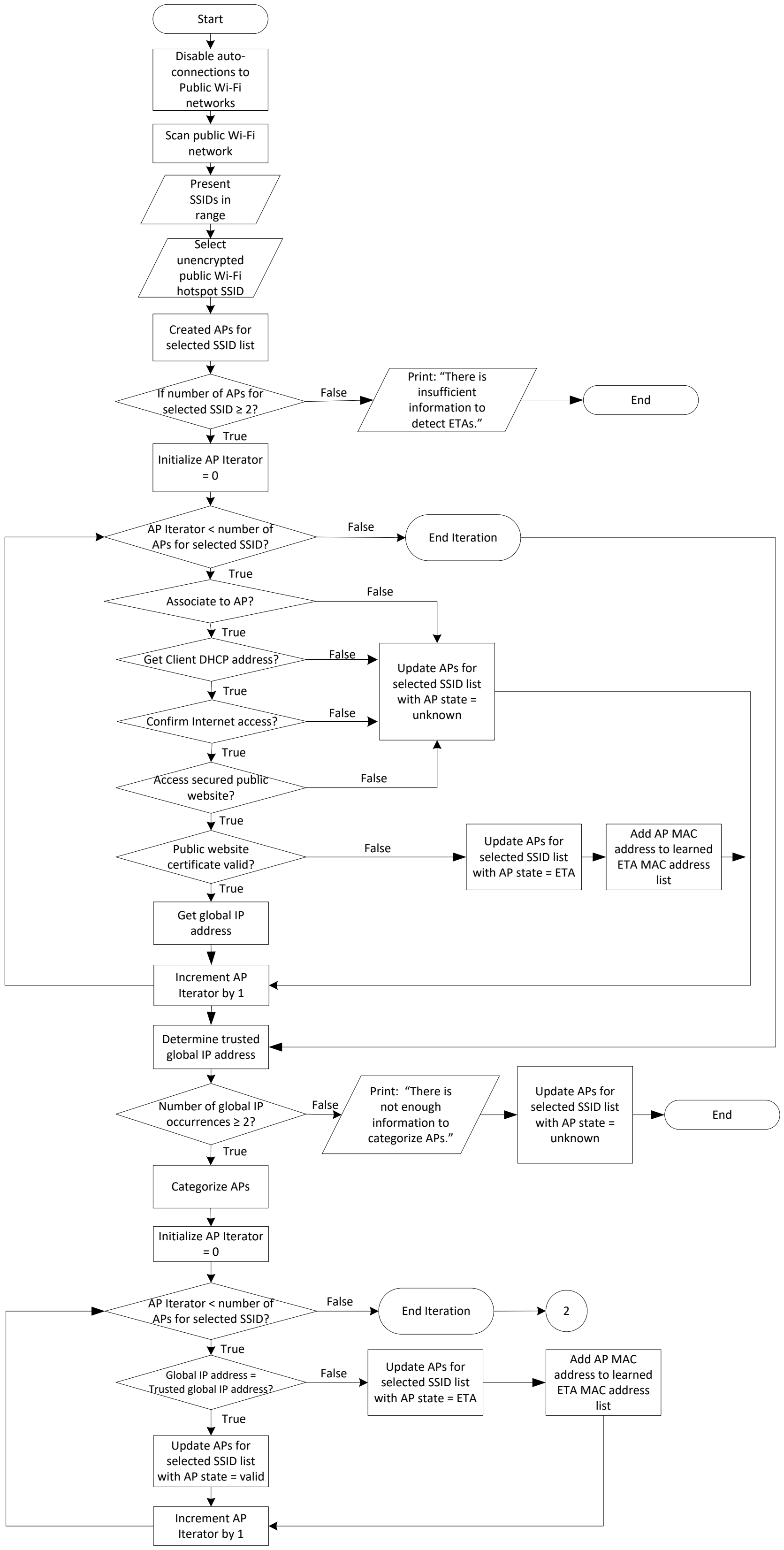

Start
Disable auto-connections to Public Wi-Fi networks
Scan public Wi-Fi network
Present SSIDs in range
Select unencrypted public Wi-Fi hotspot SSID
Created APs for selected SSID list
If number of APs for selected SSID ≥ 2?
False
Print: "There is insufficient information to detect ETAs."
End
True
Initialize AP Iterator = 0
AP Iterator < number of APs for selected SSID?
False
End Iteration
True
Associate to AP?
False
True
Get Client DHCP address?
False
Update APs for selected SSID list with AP state = unknown
True
Confirm Internet access?
False
True
Access secured public website?
False
True
Public website certificate valid?
False
Update APs for selected SSID list with AP state = ETA
Add AP MAC address to learned ETA MAC address list
True
Get global IP address
Increment AP Iterator by 1
Determine trusted global IP address
Number of global IP occurrences ≥ 2?
False
Print: "There is not enough information to categorize APs."
Update APs for selected SSID list with AP state = unknown
End
True
Categorize APs
Initialize AP Iterator = 0
AP Iterator < number of APs for selected SSID?
False
End Iteration
2
True
Global IP address = Trusted global IP address?
False
Update APs for selected SSID list with AP state = ETA
Add AP MAC address to learned ETA MAC address list
True
Update APs for selected SSID list with AP state = valid
Increment AP Iterator by 1

*Figure 11*. CSMETAD Algorithm Flow – Phase 1 Data Collection.

*Phase 2: Detection & Protection*

**Basic Flow:**

1. System rescans the public Wi-Fi network to rediscover APs with selected SSID.
2. System creates list of SSIDs in range.
3. System adds all the APs for selected SSID with signal level equal to or greater than -75 dBm to the **APs for selected SSID** list (new list).
4. System validates that number of APs for selected SSID is equal to or greater than 1. **(Alternative Flow "a")**
5. IF number of APs for selected SSID is equal to or greater than 1 THEN System retrieves learned ETA MAC addresses from **learned ETA MAC address** list. This list includes all ETA MAC addresses learned from the beginning of the program.
6. System removes learned ETA MAC addresses from **APs for selected SSID** list.
7. System validates that the **number of APs for selected SSID** is equal to or greater than 1. **(Alternative Flow "b")**
8. IF **number of APs for selected SSID** is equal to or greater than 1 THEN System starts iterating across all the APs in **APs for selected SSID** list.
9. Systems associates to the AP with the highest signal strength in the **APs for selected SSID** list. **(Alternative Flow "c")**
10. IF System is able to associate to the AP THEN System gets a Client DHCP address for the user. **(Alternative Flow "d")**
11. IF System is able to get a Client DHCP address for the user THEN System confirms access to the Internet. **(Alternative Flow "e")**

12. IF System is able to confirm access to the Internet THEN System connects to secured public website to retrieve global IP address of the AP. **(Alternative Flow “f”)**
13. IF System is able to access secured public website THEN System verifies that the public website certificate is valid. **(Alternative Flow “g”)**
14. IF the public website certificate is valid only THEN System is able to get the global IP address for the AP.
15. System validates that the global IP address for the AP is the same as the trusted global IP address. **(Alternative Flow “h”)**
16. IF the global IP address for the AP is the same as the trusted global IP THEN the System displays message: “Wi-Fi connection is safe. You are connected to a legitimate AP”.
17. System ends iterating across APs in **APs for selected SSID** list.
18. System waits for a disassociated wireless card event.
19. IF System receives a disassociated wireless card event THEN System proceeds to rescans the public Wi-Fi network. Algorithm phase 2 repeats (infinite loop).

**<u>Alternative Flows:</u>**

*a) number of APs for selected SSID is less than 1*

On step 4 of the Basic Flow:

1. IF the **number of APs for selected SSID** is less than 1 THEN
2. System displays message: “Your device is out of range for the selected public Wi-Fi hotspot. Please move closer”.
3. Flow of events returns to step 1 of the Basic Flow.

*b) number of APs for selected SSID is less than 1*

On step 7 of the Basic Flow:

1. IF the number of APs with the selected SSID is less than 1 THEN
2. System displays message: “You are located on the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi Hotspot”.
3. Flow of events returns to step 1 of the Basic Flow.

*c) system is not able to associate to an AP*

On step 9 of the Basic Flow:

1. IF System is not able to associate to an AP THEN
2. System associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
3. Flow of events returns to step 9 of the Basic Flow.

*d) system is not able to get a Client DHCP address for the user*

On step 10 of the Basic Flow:

1. IF the System is not able to get a Client DHCP address for the user THEN
2. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
3. Flow of events returns to step 9 of the Basic Flow.

*e) system is not able to confirm access to the Internet*

On step 11 of the Basic Flow:

1. IF the System is not able to confirm access to the Internet THEN
2. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.

3. Flow of events returns to step 9 of the Basic Flow.

*f) system is not able to access secured public website*

On step 12 of the Basic Flow:

1. IF the System is not able to access secured public website THEN
2. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
3. Flow of events returns to step 9 of the Basic Flow.

*g) invalid certificate*

On step 13 of the Basic Flow:

1. IF the System receives an invalid certificate message THEN
2. System has detected an ETA on the public Wi-Fi network. System displays message: "CSMETAD has detected an ETA on the public Wi-Fi hotspot".
3. System adds the AP MAC address to the **learned ETA MAC address** list (if ETA is not in the list).
4. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
5. Flow of events returns to step 9 of the Basic Flow.

*h) global IP address for an AP is not the same as the trusted global IP address*

On step 15 of the Basic Flow:

1. IF the System determines that the global IP address for an AP is not the same as the trusted global IP address THEN
2. System has detected an ETA on the public Wi-Fi network. System displays message: "CSMETAD has detected an ETA on the public Wi-Fi hotspot".

3. System adds the AP MAC address to the **learned ETA MAC address** list (if ETA is not in the list).
4. System disassociates from current AP and associates to the next AP with the highest signal strength on the **APs for selected SSID** list.
5. Flow of events returns to step 9 of the Basic Flow.

**<u>Input and Output details:</u>**

1. **APs for selected SSID** list = list of AP MAC address, signal level/RSSI, encryption status, frequency, channel, Client DHCP address, Internet access, secured public website access, certification status, global IP address, and AP state.  This list contains only the APs with encryption off and signal level equal to or greater than -75dBm.
2. **Learned ETA MAC address** list = list of ETA MAC addresses.

**<u>Rule details:</u>**

1. **Number of APs for selected SSID =** the number of APs for selected SSID must be equal to or greater than 1.

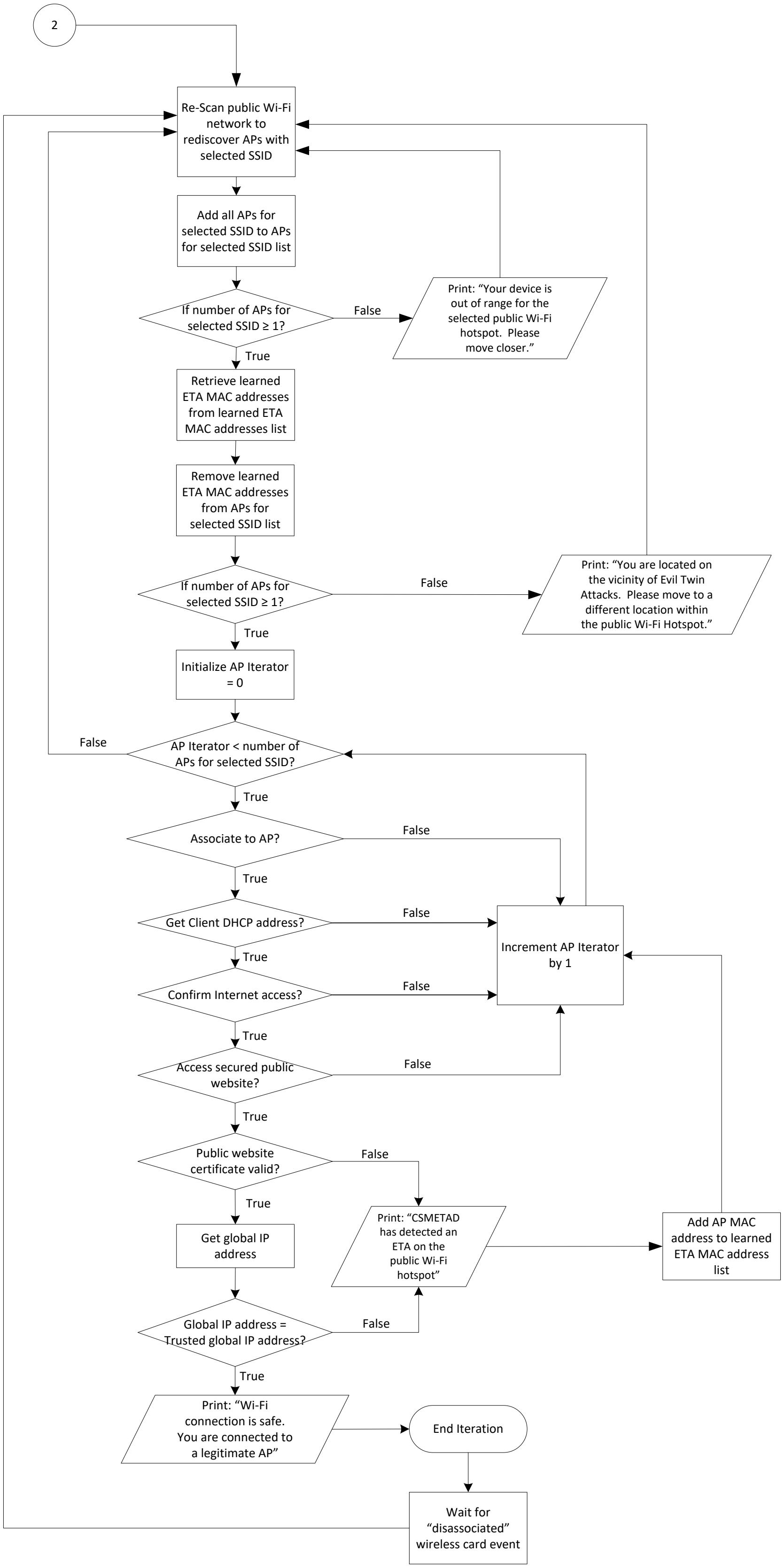

2
Re-Scan public Wi-Fi network to rediscover APs with selected SSID
Add all APs for selected SSID to APs for selected SSID list
If number of APs for selected SSID ≥ 1?
False
Print: "Your device is out of range for the selected public Wi-Fi hotspot. Please move closer."
True
Retrieve learned ETA MAC addresses from learned ETA MAC addresses list
Remove learned ETA MAC addresses from APs for selected SSID list
If number of APs for selected SSID ≥ 1?
False
Print: "You are located on the vicinity of Evil Twin Attacks. Please move to a different location within the public Wi-Fi Hotspot."
True
Initialize AP Iterator = 0
False
AP Iterator < number of APs for selected SSID?
True
Associate to AP?
False
True
Get Client DHCP address?
False
Increment AP Iterator by 1
True
Confirm Internet access?
False
True
Access secured public website?
False
True
Public website certificate valid?
False
True
Print: "CSMETAD has detected an ETA on the public Wi-Fi hotspot"
Add AP MAC address to learned ETA MAC address list
Get global IP address
Global IP address = Trusted global IP address?
False
True
Print: "Wi-Fi connection is safe. You are connected to a legitimate AP"
End Iteration
Wait for "disassociated" wireless card event

*Figure 12*. CSMETAD Algorithm Flow – Phase 2 Detection & Protection.

### 6. Step-by-step artifact testing procedures

Rigorous testing took place with the CSMETAD system based on Hevner's DSR principle 5 in order to verify that the architecture components were working effectively according to the design.

*Lab Environment Testing*

1. The TCP/IP utility "ifconfig" was used to verify the correct address configuration of the lab equipment, once the devices in the topology were connected, configured and developed in the construction phase.
2. The Cisco Operating System "show run" command was used to prove and troubleshoot the configuration of the router, switch, wireless controller, and wireless access points.
3. The TCP/IP utility "ping" was used to verify connectivity between router, switch, wireless controller, and wireless access points.
4. The TCP/IP utility "traceroute" was used to discover the path between devices across the topology.

Appendix B shows Algorithm test cases and results.

## 7. Transition client-side mobile evil twin attack detection system into production

### Artifact Production

After the construction was complete, the CSMETAD system was brought into production mode. CSMETAD initially aimed at detecting real mobile evil twin AP attacks in the wild at a hotel property. Since no mobile evil twin APs were detected in the wild during the field evaluation period, the author proceeded to evaluate the system with the mobile evil twin AP used in the lab.

### Artifact Evaluation

The client-side mobile evil twin attack detection system was evaluated based on Hevner's principle 3 that asserts that the utility, quality, and efficacy of a design artifact must be rigorously validated via well executed evaluation methods. The author extensively evaluated the performance of the client-side evil twin attack detection method by implementing a prototype system. The prototype system was evaluated in two environments. First, in a lab to analyze the requirements and demonstrate the effectiveness in a controlled environment. Second, in the field at a hotel public Wi-Fi hotspot to extensively evaluate the robustness of the system in practice. The client-side mobile evil twin attack detection system aimed at detecting real mobile evil twin AP attacks in the wild at a hotel property that provide free open public Wi-Fi. Since no real mobile evil twin AP attacks were detected in the wild during the field evaluation period, the author proceeded to evaluate the system with the mobile evil twin AP used in the lab. Similar approach was used by Hossen & Wenyuan's study and the remainder of the client-side evil twin attack detection studies referenced in this dissertation.

The client-side evil twin detection method developed as part of this dissertation was tested against Hossen & Wenyuan's (2014) method for detecting mobile evil twin attacks. The experiments aimed at showing that the detection system developed can detect mobile evil twin attacks more effectively and efficiently.

The techniques to evaluate the effectiveness and efficiency of the system were based on Hossen & Wenyuan's (2014) evaluation methodology which has been published and validated and included the following:

1. Collected data from a hotel public Wi-Fi hotspot (public spaces).
2. Ran the experiments on both weekdays and weekends for a period of 5 weeks (2 weeks at the lab and 3 weeks at the hotel).
3. Collected approximately 300 hours of data.
4. Collected more than 151,000 instances of data.
5. Ran the detection system 140 times at the lab and 210 times at a hotel public Wi-Fi hotspot.
6. Monitored the network with Wireshark packet analyzer.

For efficiency, the author used Hossen and Wenyuan's technique to measure time delay but also leveraged Nakhila et al.'s technique to improve upon Hossen and Wenyuan's. Nakhila et al. included a complete list of measurements and factors impacting efficiency.

In this study, the author used a researcher-participant approach. According to Richey and Klein (2007), researchers are often the designer/ developers. In other words, by design they "go native" and observe themselves. "The researcher who ceases to be conscious of the observer role is said to be going native" (Singleton & Straits, 2005). In this study, the author participated as the user of the client-side evil twin attack detection system and the researcher observing the client-side evil twin attack detection system performance.

# References

Abdollah. T. (2007, March 16). Ensnared on the wireless web. *Los Angeles Times*. Retrieved from http://articles.latimes.com/2007/mar/16/local/me-wifihack16

ABI Research, Global public Wi-Fi hotspots will reach 7.8 Million in 2015 and continue to grow at a CAGR of 11.2% through 2020 (2015). Retrieved from https://www.abiresearch.com/press/global-public-wi-fi-hotspots-will-reach-78-million/

Albright, L., & Malloy, T. E. (2000). Experimental validity: Brunswik, Campbell, Cronbach, and enduring issues. *Review of General Psychology, 4*(4), 337-353.

Beck, R., & Weber, S. (2013). Enhancing IT artifact construction with explanatory and predictive knowledge in design science research. *Journal of Information Technology Case & Application Research, 15*(1).

Briggs, R. O., & Schwabe, G. (2011). On expanding the scope of design science in IS research. *In Service-Oriented Perspectives in Design Science Research* (pp. 92-106). Springer Berlin Heidelberg.

CentOS, Linux CentOS7 Manuals Release Notes (2016). Retrieved from https://wiki.centos.org/Manuals/ReleaseNotes/CentOS7

Chandrasekaran, B. (1990). Design problem solving: a task analysis. *AI Magazine, 11*(4), 59.

Cheng, N., Wang, X., Cheng, W., Mohapatra, P., & Seneviratne, A. (2013). Characterizing privacy leakage of public Wi-Fi networks for users on travel. *Proceedings of the 2013 IEEE International Conference on Computer Communications (INFOCOM)*, 2769-2777.

Choi, J., Chang, S., Ko, D., & Hu, Y. (2011). Secure MAC-Layer protocol for captive portals in wireless hotspots. *Proceedings of the 2011 IEEE International Conference on Communications (ICC),* 1-5.

Cisco Systems, Cisco M10 Router User Guide (2010). Retrieved from http://downloads.linksys.com/downloads/userguide/1224655305646/Valet_Valet_Plus_M10_M20_UG_US_V10_D-WEB_3425-014530.pdf

Cisco Systems, Cisco Catalyst 3560 Series Switches Specifications (2009). Retrieved from http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3560-series-switches/product_data_sheet09186a00801f3d7d.pdf

Cisco Systems, Catalyst 3560 Switch Hardware Installation Guide (2010). Retrieved from http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3560/hardware/installation/guide/3560hig.pdf

Cisco Systems, Cisco 2500 Series Wireless Controllers Specifications (2015). Retrieved from http://www.cisco.com/c/en/us/products/collateral/wireless/2500-series-wireless-controllers/data_sheet_c78-645111.pdf

Cisco Systems, Cisco 2500 Series Wireless Controller Deployment Guide (2017). Retrieved from https://www.cisco.com/c/en/us/support/docs/wireless/2500-series-wireless-controllers/113034-2500-deploy-guide-00.html

Cisco Systems, Cisco Aironet 3500 Series Access Point Specifications (2012). Retrieved from http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1250-series/data_sheet_c78-594630.pdf

Cisco Systems, Cisco Aironet 3500 Series Lightweight Access Point Installation Guide (2014). Retrieved from http://www.cisco.com/c/en/us/td/docs/wireless/access_point/3500/quick/guide/ap3500getstart.pdf

Feng. P. (2012). Wireless LAN security issues and solutions. *IEEE Symposium on Robotics and Applications (ISRA)*, 921-924.

Google Open Source. (n.d.). Memorable site for testing clients against bad SSL configs. Retrieved from https://opensource.google.com/projects/badssl.com

Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly*, *30*(3), 611-642.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems, 8*(5), 312-335.

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2).

Habibi, A., Seyed, M., & Samadi, B. (2009). A survey on wireless security protocols (WEP, WPA and WPA2/802.11i). *IEEE 2$^{nd}$ International Conference on Computer Science and Information Technology (ICCSIT)*, 48-52.

Han, H., Sheng, B., Tan, C. C., Li, Q., & Lu, S. (2009). A measurement based rogue AP detection scheme. *Proceedings of the 2009 28th Annual Conference of the IEEE Communications Society*, 1593-1601.

Han, H., Sheng, B., Tan, C. C., Li, Q., & Lu, S. (2011). A timing-based scheme for rogue AP detection. *IEEE Transactions on Parallel and Distributed Systems, 22*(11), 1912-1925.

Harris Poll, Are you protected from hackers on Public Wi-Fi? (2014). Retrieved from http://blog.privatewifi.com/are-you-protected-from-hackers-on-public-wifi-infographic/

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems, 19*(2), 87-92.

Hossen, M., & Wenyuan, X. (2014). CETAD: Detecting evil twin access point attacks in wireless hotspots. *Proceedings of the 2014 IEEE Conference on Communications and Network Security*, 238-246.

Hostapd, Hostapd: IEEE 802.11 AP (2013). Retrieved from https://w1.fi/hostapd/

Hsu, F., Wang, C., Hsu, Y., Cheng, Y., & Hsneh, Y. (2015). A client-side detection mechanism for evil twins. *Computers and Electrical Engineering.*

IDC, Worldwide Enterprise WLAN market sees stronger growth in second quarter of 2017 as wireless digital transformation continues, according to IDC (2017). Retrieved from https://www.idc.com/getdoc.jsp?containerId=prUS43065517

JiWire, Mobile Audience Insights Report (2013). Retrieved from http://www.ninthdecimal.com/wp-content/uploads/2014/02/JiWire_Insights_Q4_2013.pdf

Kali, Kali Linux 2018.1 Release Notes (2018). Retrieved from https://www.kali.org/news/kali-linux-2018-1-release/

Kali, Kali Linux 4.14.0 Installing Kali Linux (2018). Retrieved from https://docs.kali.org/category/installation

Kelley, B. (2012). *American Generation Y and The Hotel of 2030* (Doctoral dissertation). Retrieved from Digital Scholarship@UNLV (Order 1470).

Kelly, K. (2014, January 7). New IEEE 802.11ac specification driven by evolving market need for higher multi-user throughput in wireless LANs. *IEEE Standards Association*. Retrieved from https://web.archive.org/web/20140112011626/http://standards.ieee.org/news/2014/ieee_802_11ac_ballot.html

King, W. R., & He, J. (2005). External validity in IS survey research. *Communications of the Association for Information Systems, 16,* 880-894.

Kim, T., Park, H., Jung, H., & Lee, H. (2012). Online detection of fake access points using received signal strengths. *Proceedings of the 2012 IEEE 75th Vehicular Technology Conference*, 1-5.

Kirankumar, B., Babu, V. M., Prasad, D. S., & Vishnumurthy, R. (2012). Wireless Security System. *International Journal of Computer Science and Information Security, 10*(4), 140-144.

Kumar, A., & Paul, P. (2016). Security analysis and implementation of a simple method for prevention and detection against evil twin attack in IEEE 802.11 wireless LAN. *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, 176-181.

Lanze, F., Ponce-Alcaide, I., Panchenko, A., & Engel, T. (2014). Undesired relatives: protection mechanisms against the evil twin attack in IEEE 802.11. *Proceedings of the 10th ACM symposium on QoS and security for wireless and mobile networks,* 87-94.

Leedy, P. D., & Ormrod, J. E. (2005). *Practical research: planning and design (8th ed.)*. Upper Saddle River, NJ: Prentice Hall.

Lenovo, Lenovo Thinkpad X1 Carbon Specifications (2013). Retrieved from https://support.lenovo.com/mn/en/solutions/pd027202

Lenovo, Lenovo Thinkpad X1 Carbon Specifications (2016). Retrieved from https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-x/Thinkpad-X1-Carbon-4th-Gen/p/22TP2TXX14G

March, S. T, & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems, 15*(4), 251-266.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *MIS Quarterly, 26*(3), 179-212.

Monica, D., & Ribeiro, C. (2011). WiFiHop - mitigating the evil twin attack through multi-hop detection. *Proceedings of the 16th European conference on Research in computer security,* 21-39.

Motorola, Moto e$^5$ specifications (2018). Retrieved from https://www.motorola.com/us/products/moto-e-plus-gen-5

Motorola, Tethered Modem Connection - moto e5 play. Retrieved from https://www.verizonwireless.com/support/knowledge-base-217411/

Myslewski, R. (2011, April 15). Wireless devices to break one-billion barrier in 2011. *The Register*. Retrieved from http://www.theregister.co.uk/2011/04/15/wireless_projections/

Nakhila, O., Dondyk, E., Amjad, M. F., & Zou, C. (2015). User-side Wi-Fi evil twin attack detection using SSL/TCL protocols. *Proceedings of the 2015 IEEE 12th Consumer Communications and Networking Conference (CCNC)*, 1-6.

NetBeans, NetBeans IDE 8.1 Release Notes (2015). Retrieved from https://netbeans.org/community/releases/81/relnotes.html

NetBeans, NetBeans IDE 8.1 Installation Instructions (2015). Retrieved from https://netbeans.org/community/releases/81/install.html

Nikbakhsh, S., Zamani, M., Abdul Manaf, A. B., & Janbeglou, M. (2012). A novel approach for rogue access point detection on the client-side. *Proceedings of the 2012 IEEE 26th International Conference on Advanced Information Networking and Applications Workshops*, 684-687.

NIST, Guide to IPsec VPNs (2005). Retrieved from http://csrc.nist.gov/publications/nistpubs/800-77/sp800-77.pdf

NIST, Guide to securing legacy IEEE 802.11 wireless networks (2008). Retrieved from http://www.nist.gov/customcf/get_pdf.cfm?pub_id=890006

Norton by Symantec, Norton Cybercrime Report (2011). Retrieved from https://www.symantec.com/content/de/de/about/downloads/PressCenter/NCR_Global_Fact_Sheet.pdf

Oracle, The Java Language Specification Java SE 8 Edition (2015). Retrieved from https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf

Oracle, Java Platform Standard Edition Installation Guide (2016). Retrieved from http://docs.oracle.com/javase/8/docs/technotes/guides/install/index.html

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2008). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45-78.

Private Wi-Fi, How Wi-Fi hacks occur (2011). Retrieved from http://blog.privatewifi.com/how-wifi-hotspot-hacks-occur/

Private Wi-Fi, 76% say free Wi-Fi can lead to identity theft (2013). Retrieved from http://blog.privatewifi.com/infographic-76-say-free-wifi-can-lead-to-identity-theft/

Richey, R. C., & Klein, J. D. (2007). *Design and Development Research*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Singh, P., Mishra, M., & Barwal., P. N. (2014). Analysis of security issues and their solutions in wireless LAN. *IEEE 2014 International Conference on Information Communication and Embedded Systems (ICICES),* 1-6.

Singleton, R. A., & Straits, B. C. (2005). *Approaches to social research (5th ed.).* New York: Oxford University Press.

Song, Y., Yang, C., & Gu, G. (2010). Who is peeping at your passwords at Starbucks? - To catch an evil twin access point. *Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN),* 323-332.

Szongott, C., Henne, B., & Smith, M. (2012). Mobile evil twin malnets – the worst of both worlds. *Springer Cryptology and Network Security. Proceedings of the 2012 11th International Conference (CANS),* 126-141.

Szongott, C., Brenner, M., & Smith, M. (2015). METDS – A self-contained, context-based detection system for evil twin access points. *Springer Financial Cryptography and Data Security, 19th International Conference, 8975,* 370-386.

The Guardian, Wi-Fi security flaw for smartphones puts your credit card at risk (2011). Retrieved from http://www.theguardian.com/technology/2011/apr/25/wifi-security-flaw-smartphones-risk

Venable, J. (2006). The role of theory and theorizing in design science research. In *Proceedings of the 1st International Conference on Design Science in Information Systems and Technology (DESRIST),* 1-18.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research, 3*(1), 36-59.

Wireshark, Wireshark 2.0.0 Release Notes (2014). Retrieved from https://www.wireshark.org/docs/relnotes/wireshark-2.0.0.html

Wireshark, Wireshark User’s Guide (2014). Retrieved from https://www.wireshark.org/docs/wsug_html/

Yang, C., Song, Y., & Gu, G. (2012). Active user-side evil twin access point detection using statistical techniques. *IEEE Transactions on Information Forensics and Security, 7*(5), 1638-1651.