2016-11-01

Towards Using Certificate-Based Authentication as a Defense Against Evil Twins in 802.11 Networks

Travis S. Hendershot
Brigham Young University

Towards Using Certificate-Based Authentication as a Defense Against Evil Twins in 802.11 Networks

Travis S. Hendershot

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Kent Seamons, Chair
Daniel Zappala
Mark Clement

Department of Computer Science

Brigham Young University

ABSTRACT

Towards Using Certificate-Based Authentication as a Defense Against
Evil Twins in 802.11 Networks

Travis S. Hendershot
Department of Computer Science, BYU
Master of Science

Wireless clients are vulnerable to exploitation by evil twins due to flaws in the authentication process of 802.11 Wi-Fi networks. Current certificate-based wireless authentication protocols present a potential solution, but are limited in their ability to provide a secure and usable platform for certificate validation. Our work seeks to mitigate these limitations by exploring a client-side strategy for utilizing alternative trust models in wireless network authentication. We compile a taxonomy of various trust models for conducting certificate-based authentication of wireless networks and methodically evaluate each model according to desirable properties of security, usability, and deployability. We then build a platform for leveraging alternative certificate-based trust models in wireless networks, present a proof-of-concept using one of the most promising alternative validation models identified—a whitelisting and pinning hybrid—and examine its effectiveness at defending against evil twin attacks in 802.11 networks.

Keywords: Wireless networks, authentication, public key cryptography, evil twin

ACKNOWLEDGMENTS

Thanks to Dr. Kent Seamons for his countless hours of advisement that made this work possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today's widespread deployment of Wi-Fi networks enables users to maintain a near-constant connection to the Internet. The Wi-Fi Alliance recently reported the remarkable existence of over 47 million public Wi-Fi hotspots worldwide [25]. In addition, an estimated 800 million households world-wide will maintain Wi-Fi access in 2016 [15]. Users are increasingly relying on Wi-Fi networks to connect smart homes and mobile devices such as smartphones, tablets, and laptops. It is therefore essential that the community considers the sensitivity of the information being wirelessly transmitted over these networks.

Despite considerable attention in the past decade, a majority of networks worldwide remain open to an attack from an evil twin access point. This attack, which was brought into public focus due to its popular disclosure by AirDefense at the RSA Conference of 2007 [10], still persists as a fundamental vulnerability in wireless networks.

The most widespread wireless authentication protocol today, WPA2-PSK, is vulnerable to this attack if a network is configured with an insecure password or the network adopts the current practice of publishing pre-shared keys for public Wi-Fi use. Networks using the 802.1X authentication protocol can also be vulnerable to an evil twin attack when client Wi-Fi supplicants are not set up to correctly validate network certificates. Furthermore, any Wi-Fi access point left in an open configuration can be trivially exploited, as well.

There are two standard approaches to defending against evil twins: detection techniques and prevention strategies. The vast majority of the literature exploring evil twin defenses is centered around detection techniques. A variety of methods exist for identifying evil twins,

but they are limited in their applicability and often cannot detect all variations of an evil twin. Prevention strategies focus on certificate-based authentication, but rely on the existing Certificate Authority infrastructure. This places additional costs on networks and inherits all the flaws of the CA system [1, 5, 19, 22]. In addition, certificate-based authentication typically validates the name of an associated RADIUS server, not the SSID of the wireless network, which causes additional complications.

To provide stronger protection against evil twin attacks, we propose the use of enhanced certificate-based authentication, with alternative trust models that address weaknesses of the CA model and current practices. We first conduct a comparative analysis of alternative trust models in the scope of wireless network authentication, showing that these models have the potential to provide increased security, usability, and deployability benefits that can foster the growth of certificate-based authentication in wireless networks. We then implement a platform to provide flexibility for utilizing alternative certificate trust models within wireless networking clients. This platform leverages the TrustBase middleware [18], a tool used in ongoing certificate-based research on the Internet that provides pluggable modules for alternative certificate validation models. We present a proof-of-concept implementation of a whitelisting and pinning hybrid trust model within this platform, and analyze its benefits of preventing a variety of evil twin attacks.

In this work, we make the following contributions:

1. We propose the use of enhanced certificate-based authentication with alternative trust models as a viable strategy for preventing exploitation by evil twins.
2. We present a taxonomy and comparative analysis of public key certificate trust models for wireless network authentication. We describe the benefits of merging individual trust models into more comprehensive hybrid models that leverage their individual strengths to overcome weaknesses.
3. We extend an open source *wpa_supplicant* to use certificate-based authentication with alternative trust models through an integration of the TrustBase authentication system.

Our platform extends the scope of TrustBase to include wireless networks. This result has the potential to unify certificate-based authentication in the web and wireless domains.

4. We prototype a hybrid trust model based on our comparative analysis that combines whitelisting and pinning. We analyze its ability to combat the threat model of an evil twin. Our implementation demonstrates how researchers can leverage TrustBase to study certificate-based authentication for wireless networks.

Our results indicate that there are clear and immediate benefits to encouraging the use of certificate-based validation of wireless networks. However, the current practice of relying on Certificate Authorities restricts the security, usability, and deployability of using certificates to authenticate wireless networks. By utilizing trust models that fit the appropriate needs of varying network environments, such as those found in homes or public locations, certificate-based authentication can adapt to the specific environments where it will be needed. These alternative trust models present many unique benefits over the current status quo, and thus assist in overcoming hurdles to mainstream adoption. Furthermore, by identifying a client-side strategy that enables flexible certificate validation without the need for major protocol changes across a network, this solution can be rolled out gradually to client devices without the need to immediately reconfigure every network for compatibility.

Chapter 2

Related Work

The community has explored many diverse solutions to evil twin attacks, each with their own advantages and disadvantages. A few basic but popular solutions include enforcing the HTTPS protocol on all wireless web traffic, incorporating a local firewall and antivirus software, or using a Virtual Private Network (VPN). These solutions may be effective to various degrees but are incomplete. For example, using either HTTPS or a VPN does not secure the use of local network resources, and these solutions place additional cost/delay on network traffic. Therefore, a number of advanced techniques have been proposed to compensate for the lack of network authentication by wireless clients.

Techniques developed for detecting the presence of evil twins can be divided into two broad categories, depending on whether they are implemented on the network side or the client side. The following two sections take a deeper look into these two categories. We then examine previous work on certificate-based authentication within wireless networking, and follow with related work on alternative trust models for validating public key certificates within TLS on the Internet.

2.1 Network-side Evil Twin Detection

Network-side solutions provide a method for network administrators to monitor their networks and detect the presence of any unauthorized access points. These solutions can be further subdivided by whether they are implemented as a wired or wireless solution.

Wired: One class of techniques for detecting a rogue access point is *wired traffic fingerprinting*. This approach first attempts to characterize the typical traffic of a network. Then, later on, when atypical traffic is discovered, the network can flag the possibility that a rogue access point has been connected. Methods of fingerprinting include the analysis of packet interarrival times, TCP ACK arrival times, and packet round-trip times [2].

A second but similar technique is *wired device fingerprinting*. In this technique, a machine on the network queries connected devices in order to fingerprint their unique characteristics [2]. This assumes that each enterprise access point will be a different device model than the attacker's evil twin device. When the network queries a connected evil twin device, the response will show that the evil twin is an outlier and should not be allowed on the network. This method fails if the attacker either uses a device similar to the others on the network, or spoofs the use of a similar device by manipulating the query's response.

Wired approaches often have the disadvantage of being unable to detect the presence of evil twins who hide behind a legitimate access point (i.e., those that forward network traffic from their clients to another AP on the network.) Additionally, advanced attackers can exploit the fact that many wired techniques utilize statistical analyses by countering with traffic-shaping techniques to blend in their traffic with the rest of the packets on the network. In these cases, a wireless approach to identifying evil twins may be more effective.

Wireless: The most common wireless technique is *wireless passive sniffing*. This consists of deploying sensors throughout the network and discovering both physical- and link-layer information from devices that are propagating Wi-Fi signals through the air. Information helpful for discovering an evil twin includes the media-access-control (MAC) addresses, signal strengths, RF measurements, and access point control messages of nearby devices [2]. Any anomalies in these measurements can signal that an unexpected wireless device is being used within the sampled region. This method can be effective at proactively identifying the presence of rogue access points, but requires additional hardware that can be very expensive (one Laptop Analyzer by Air Magnet costs around US$3,000) [2]. Thus, this

solution is not easily scalable. It also fails to detect attackers that utilize directional antennas or other methods to decrease the ability to sniff their traffic.

Another technique called *wireless active fingerprinting* relies on actively probing local Wi-Fi stations and analyzing their responses to a certain set of well-crafted requests. This technique, proposed by Bratus et al., can discover certain qualifying information about the hardware, firmware, and device drivers by classifying the results that come back from these custom requests [3]. This classification can overcome the use of MAC address spoofing in evil twins and can notify the administrator about new devices that do not match the typical expectations of routers on the network. However, many attackers are sophisticated and will not respond to active probing—thus reducing the value of this approach.

All of these network-side approaches fail to identify evil twins that are not co-located in the vicinity of the actual network.

2.2 Client-side Evil Twin Detection

From the client side, there are also a variety of approaches for detecting the presence of an evil twin.

Fingerprinting: Some methods attempt to identify a difference in the fingerprint of the network. A branch of techniques called *multi-hop detection* attempts to determine whether an evil twin is introducing an additional "hop" into the wireless network stream. For instance, Han et al. measure the rount-trip time (RTT) between the client and the local Domain Name System (DNS) server in order to determine whether the RTT is significantly longer than normal [9]. These techniques assume that the evil twin is forwarding received packets on to the legitimate access point. Lanze et al. utilize the dependency of *clock skews* on temperature within wireless access points in order to perform remote device fingerprinting of APs [12]. Clocks within wireless routers vary at small yet observable differences, which allows the clock to be used as a unique identifier for a device. This strategy is effective but requires the client to have previously measured the clock skew of the actual device and

does not scale to networks that use multiple access points. In a separate paper, Lanze et al. describe the process of using radiometric signal properties to fingerprint a device [13]. This technique can be effective but requires dedicated specialized hardware in order to be carried out. Active fingerprinting may also be accomplished by sending out probe requests, similar to the probing technique used by a network.

Location Tracking: *Context Leashing* is a method for detecting evil twins that involves remembering the context of available Wi-Fi networks found locally to the network [8]. When a user connects to a wireless network for the first time, their device remembers the other wireless networks available at that location. Then, when the user attempts to reconnect to the network, the device verifies that the same local wireless networks are still available. This strategy only works when the user has previously connected to the network, and it is unable to identify situations where a legitimate change in available networks has occurred (such as a neighbor adding or removing a network.) It also only works when the evil twin is not co-located with the actual network.

Software-based AP: Since one of the easiest ways to set up an evil twin is through the use of a software-based access point, as opposed to a dedicated router, a variety of techniques center around detecting anomalies in software-based APs. These techniques measure either accuracy flaws due to router emulation or abnormalities resulting from the client's networking hardware [14]. This relies on the assumption that all evil twins are software-based, and all software-based APs are evil twins, which may not be universally true.

2.3 Wireless Network Authentication

A few Extensible Authentication Protocol (EAP) modules have been developed in order to enable certificate-based authentication in wireless networks. EAP-TLS is an IETF open standard that uses a TLS handshake to perform mutual authentication [21]. Though not mandated by the standard, most implementations require that both the supplicant and the authentication server present certificates during the handshake. EAP-TTLS is another IETF

open standard that typically uses a certificate only for server authentication and tunnels another EAP module for client authentication [7]. Both of these TLS-based EAP modules require the use of a CA-signed certificate, according to their specifications. EAP-PEAP is an additional EAP module that has been developed in a joint effort by Microsoft, Cisco Systems, and RSA Security [26]. It is similar to EAP-TTLS in that it uses a certificate for server authentication and encapsulates another protocol, EAP-MSCHAPv2, for client authentication.

Gonzales et al. proposed EAP-SWAT, a new EAP module [8]. This module would be similar to EAP-TTLS, except that instead of relying on the CA model for validation, it would require the client to trust the certificate in a "trust-on-first-use" manner. To the best of our knowledge, no prototypes of this proposal were ever built.

A joint effort by Byrd et al. experimented with providing certificate-based authentication in open wireless networks, referred to as Secure Open Wireless Networking [4]. This work seeks to enable EAP-TLS without the dual requirement for a client certificate. The motivation is that by using EAP-TLS for server-side only authentication, networks can achieve an authentication handshake similar to TLS on the Internet.

Ou presents a stopgap solution using EAP-PEAP with dummy MS-CHAPv2 credentials in order to provide network authentication and session key establishment on an "open" public Wi-Fi network [20].

2.4 Certificate Authentication on the Web

X.509 certificates are used when authenticating web servers via the Hypertext Transfer Protocol over TLS (HTTPS). Much of the literature contends that the Certificate Authority system used for validating certificates over HTTPS has significant weaknesses.

Clark and van Oorschot claim that there is currently a disintegration of confidence in the ability of the Certificate Authority system to support the HTTPS certificate infrastructure [5]. They cite a loosening in issuance requirements, an abundancy of CAs, and examples

of CA compromises all as issues leading to elevated concern within the security community about the efficacy of the CA model to prevent man-in-the-middle (MitM) attacks on HTTPS. They also expand upon the CA trust model by identifying other techniques that could provide heightened security to HTTPS. A few techniques they identify are key pinning, multipath probing, and whitelisting.

Bates et al. further examine growing doubt in the ability of the current CA model to verify web server certificates. They build a client tool called CertShim [1] that enables alternative trust models for verifying TLS traffic on the web. Oppliger calls for certificate legitimation via an evolution of the current CA trust model after showing that various attacks against trusted CAs such as Comodo and DigiNotar have eroded the legitimacy of certificate validation on the Web [19]. Soghoian and Stamm examine cases where the CA system has been compromised by nation-states who have compelled trusted CAs to grant invalid certificates for the purpose of government interception [22]. Wang et al. propose using the CA model but requiring multiple CA signatures for a certificate [23].

Various implementations of alternative trust models have been explored as candidates for expanding or replacing the CA system. Certificate Patrol[1] is a Firefox plugin that implements certificate pinning on the web. DNS-based Authentication of Named Entities (DANE) is a key pinning strategy wherein servers pin their public key in their associated DNSSEC record [11]. Perspectives is a validation strategy proposed by Wendlandt et al. that enables the client to utilize multi-path probing by asking several notaries for their perspective on the proper certificate for a given web domain [24]. Convergence is an extension of Perspectives, developed by Moxie Marlinspike, that uses multi-path probing but also allows for flexible decision-making on reaching a final verdict for the certificate validation [17]. Certificate Transparency is an IETF open standard that utilizes logs, monitors, and auditors to track valid certificates for servers on the web [16]. The Monkeysphere[2] project uses OpenPGP's web of trust model to validate server certificates; web of trust is a decentralized

[1] https://addons.mozilla.org/en-US/firefox/addon/certificate-patrol/
[2] http://web.monkeysphere.info/

method where users in the community sign the public keys of other users, thus endorsing that user as the key's owner. We consider many of these same trust models in the context of wireless network authentication.

Chapter 3

Wireless Security Protocols

The most prevalent wireless network security protocol in use today is Wi-Fi Protected Access 2 (WPA2). This protocol, based on the latest IEEE 802.11i standard, utilizes a four-way handshake to prove that both the client and the network have mutual knowledge of a pre-shared key (PSK). If this handshake fails, the connection is terminated. If it is successful, the client station (*supplicant*) derives a Pairwise Transient Key (PTK) in conjunction with the access point (AP), or *authenticator*, of the network. The PTK is then used to provide confidentiality and integrity throughout the session via the application of one of two protocols: TKIP[1] or CCMP[2].

In some cases, the limited amount of security gained through the use of a universally shared secret in WPA2-PSK is insufficient. The entire security of this system relies on the secrecy of the pre-shared key. If the key ends up in the wrong hands, all authentication and encryption of the network is compromised. For networks that require tighter security guarantees than are given through the use of WPA2-PSK, an extension to WPA2 has been developed called 802.1X. The IEEE 802.1X standard provides additional authentication methods through the encapsulation of the Extensible Authentication Protocol (EAP) over LAN, referred to as EAPOL. This extension to the typical four-way handshake is implemented through the use of a separate *authentication server* on the LAN, called a RADIUS server. During the initial connection from the supplicant to the network access point, the authentication request from the client is forwarded to the RADIUS server, which handles

[1]Temporal Key Integrity Protocol
[2]Counter Mode Cipher Block Chaining Message Authentication Code Protocol

the steps for authentication of both parties as well as the granting of access privileges to the network. The RADIUS server has a variety of predesignated authentication methods built-in, and the network administrator is required to configure which method is acceptable for their network.

In design, the 802.1X standard is meant to provide a reliable means of mutual authentication of both the supplicant and the network. However, many networks avoid these additional security features entirely. Even networks that do integrate a RADIUS server often take shortcuts that result in undermining the overall security benefits that these additions are meant to provide in the first place. A core issue here is the lack of ease in deployment.

One complication that causes difficulty with 802.1X deployments is the additional setup required to integrate EAP-based authentication into a network connection's initial handshake. In particular, many EAP authentication methods require the use of an X.509 public key certificate that has been signed and authorized by a trusted Certificate Authority (CA) in order for successful client-side authentication of the network [7, 21, 26].

An X.509 certificate is a common authentication mechanism when two unfamiliar parties without a shared secret attempt to communicate securely over an insecure channel, such as on the Internet. The certificate typically contains the name of an issuer, the Common Name (and possibly other identifying information) of the subject being vetted, and the digital signature of the issuer verifying that this subject is authorized to act in said name's behalf. Obtaining a certificate requires the subject to either purchase it from a universally trusted CA, or set up their own Public Key Infrastructure (PKI) and distribute their PKI root authority's signing certificate to all connected client devices.

The CA trust model thus places additional cost on the network administrator and constrains the clients to reliance on a Certificate Authority as a Trusted Third-Party (TTP), rather than permitting clients the flexibility to determine their own trust relationships. Later in this paper, we will more closely examine the costs of utilizing Cerificate Authorities for

wireless network authentication and also look at the benefits of alternative approaches to certificate validation.

Chapter 4

Evil Twin Threat Model

In order to understand the implications of an evil twin attack and properly design an effective defense, it is first pertinent to analyze the attacker's motivations and the corresponding threat model.

4.1 Context

There are multiple scenarios by which an implementation of an 802.11 Wi-Fi network can lack proper network authentication. If clients do not securely authenticate their wireless networks, the door is left open for malicious actors to masquerade as a trusted network. They exploit the vulnerability by setting up a clone of the network that looks identical to the original AP. The clone is typically configured with the same name (referred to as the Service Set Identifier, or SSID) and the same protocol as the actual network, and the client's wireless supplicant cannot tell the difference between them.

This type of attack (shown in Figure 4.1) is called an *evil twin attack* because the network is set up to look like an exact twin of the original. The attack is a subcategory of an overarching brand of attacks called *rogue access points*. In an evil twin attack, the malicious actor attempts to exploit a client's trust in a familiar network in order to illicitly gain access to sensitive information or systems. Wireless clients placed in this situation put unfounded trust in their wireless network connection. Evil actors then have the means and desire to exploit this misplaced trust.

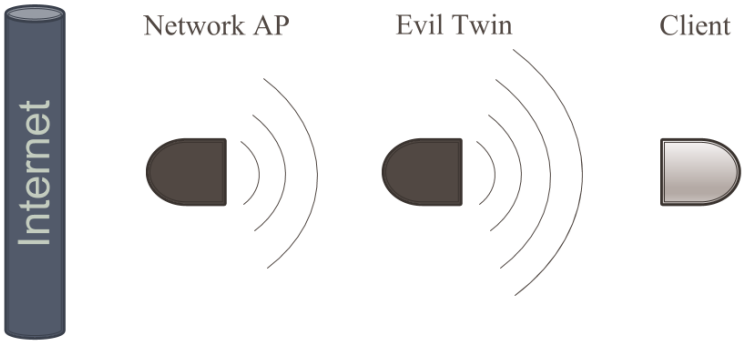

Figure 4.1: Evil Twin Attack

Without a secure method to authenticate wireless networks, clients are left vulnerable to exploitation by rogue access points. The attack scenario presented here is relevant to all users of 802.11 Wi-Fi networks who do not properly authenticate their network connections. Therefore, this is an extreme point of concern as wireless network usage continues to grow.

4.2 Evil Twin Motivations

An evil twin access point injects the attacker into the middle of network communications to and from a victim. Becoming a man-in-the-middle (MitM) to a victim's Internet traffic grants the attacker the potential to hijack the victim's session, steal their identity, or read and alter traffic that has not been sufficiently encrypted. Even in cases where end-to-end encryption is used, if the attacker can convince the victim that a spoofed X.509 certificate is valid for the endpoint they are trying to connect to, then the attacker can leverage that certificate to become a MitM within the encrypted communication. Attackers may also exploit the practice that some wireless networks take of asking the user to install a new trusted root certificate, and users are not well positioned to evaluate whether to trust the certificate being presented. Additionally, attackers may search for vulnerabilities on the victim's device, fingerprint traffic and open TCP/UDP ports on the device, or impersonate network resources such as DNS,

SMTP, DHCP, and networked printers. Worst of all, the victims are often left completely unaware that they have been compromised.

These abilities can lead to an attacker obtaining money, stealing corporate secrets, or accessing restricted resources.

4.3 Vulnerable Protocols

Clients of open networks, WPA2-PSK networks, and even WPA2-Enterprise networks with 802.1X authentication can be vulnerable to an evil twin attack.

In open networks, the requirements for setup are trivial. The attacker merely needs to set the Service Set Identifier (SSID) of their evil twin to the name of the network being cloned. Since no passwords or other means of identification are present, they will appear to the user as another legitimate AP for the cloned network.

WPA2-PSK networks require a bit more work in order to clone, since the attacker will need to acquire the pre-shared key. If the key is a secure password and maintained confidentially, this can be difficult to achieve. However, numerous public locations have adopted the habit of widely publishing their pre-shared key in order to make their Wi-Fi more available, and this sharing behavior enables attackers to set up an evil twin with minimal effort. Alternatively, if the password used is a default or relatively insecure password, then the attacker may have a chance at guessing the pre-shared key. In either case, once the pre-shared key has been obtained, the attacker needs only to set the SSID and PSK to match the legitimate AP.

WPA2-Enterprise's 802.1X authentication protocol does have the potential to thwart evil twin attackers, if network authentication is performed correctly. However, insufficient steps taken during the authentication handshake can still lead to evil twins being able to masquerade as these networks. For instance, any client supplicant that disregards the authentication server's certificate, or merely validates that the provided certificate is signed

by a trusted CA, without verifying the name of the associated RADIUS server, can be tricked into trusting an evil twin.

This weakness is compounded by the fact that some operating systems, such as Android 4.4, do not even present an option for validating the Common Name of the certificate. Worse yet, some organizations encourage users to ingnore validating certificates because network administrators have elected not to set up a valid CA-signed certificate for the network. For example, published guidelines to connect to the Wi-Fi networks of Georgetown University[1], Cornell University[2], and the University of Cambridge[3] all instruct users to leave the configuration for the root CA Certificate as "Unspecified," effectively turning off network authentication. Even though the ability to prevent evil twins exists through the use of 802.1X, it is hard in practice for users to correctly set up their certificate validation. We thus identify the need to strengthen the overall usability and deployability of this authentication protocol, which should be used as a significant tool in preventing evil twins.

4.4 Variants

There are four variations of an evil twin, as proposed by Lanze et al. [13]:

Replacement Evil Twin

A real AP is removed and replaced by an evil twin in the same location as the old one.

Coexistent Evil Twin

Both an evil twin and the real AP exist at the same location at the same time.

Remote Evil Twin

An evil twin is located at a different location than the actual network.

[1]https://uis.georgetown.edu/internet/wireless/saxanet/android
[2]http://www.it.cornell.edu/services/wifi/connectandroid.cfm
[3]http://www.ucs.cam.ac.uk/mobiledevices/android/eduroam-android

Ad-hoc Evil Twin

An evil twin is created by an attacker on-the-fly in direct response to a network probe request originating from the victim's device, irrespective of location.

Each of these four attack scenarios should be considered independently when designing a defense against evil twins, as they each present distinct differences that may alter the effectiveness of a defense strategy.

4.5 Defenses

Multiple detection strategies exist that attempt to discover and react to the presence of evil twins, rather than proactively inhibiting their ability to fool network clients. Other strategies, aimed towards prevention of attacks, attempt to design a connection handshake that is resilient to evil twins by including additional steps for authenticating a network's access point. Our work takes this latter approach.

Chapter 5

Certificate-based Wireless Network Authentication

Our solution to the evil twin vulnerability in 802.11 networks is to utilize certificate-based authentication as an effective means of removing the attacker's ability to spoof a trusted network.

Public key cryptography is common in many distributed systems that require a secure method to authenticate remote parties and ensure privacy of sensitive data (e.g., HTTPS, SSH, S/MIME). These systems rely on the effective distribution of X.509 public key certificates. Each certificate includes identifying information about the owner, along with their associated public key that can be used to establish secure communication [6].

Interestingly, there are striking similarities between the paradigm for secure web browsing over HTTPS (where the use of public key cryptography is commonplace) and that of Wi-Fi network hopping. The client is attempting to communicate with an endpoint that must be authenticated. The client has a limited amount of information (i.e., domain name, SSID) about the endpoint, and this data is not sufficient alone to securely authenticate the target. In both of these scenarios, public key certificates can be utilized to ensure proper authentication of the target.

However, despite the ability of the 802.1X protocol to support public key certificates for wireless network authentication, very few networks utilize this feature in practice. There are a number of possible factors that have led to this lack of utilization. In this chapter, we explore these factors and look at possible mitigations through the use of alternative certificate validation models.

5.1 Current State and Limitations

As defined in current standards [6], when a network administrator wants to enable certificate-based authentication, the administrator first needs to obtain a public key certificate that has been signed by a Certificate Authority. This certificate's *Common Name* is assigned according to the domain name of the RADIUS server (not the SSID of the corresponding network.) Clients then validating this certificate will need to 1) recognize the signing CA as a trusted party, and 2) either manually set the name of the RADIUS server to be verified against or be willing to accept *any* certificate that has been signed by this trusted CA. This process both burdens the network administrator and makes it difficult for the client to securely validate the network.

Additionally, there are security concerns regarding the CA system. The system restricts the ability for clients to build their own trust relationships. Instead, it requires each device to place complete trust in the predetermined Certificate Authorities. If any single CA is compromised by attackers, the entire CA system is effectively broken (since a compromised CA can sign certificates for every domain.) The ownership of CAs is also a controversial subject—some CAs are owned by nation-states, which effectively provides those states the ability to MitM wireless network traffic.

The CA system also restricts the ability to use certificate-based authentication in personal networks, where the use of self-signed certificates would be of great benefit.

5.2 Trust Model Taxonomy

A range of potential trust models could be used for certificate authentication in wireless networks. We identify these models and assess their strengths and weaknesses with regards to a set of desirable properties of security, usability, and deployability. We define properties of *security* as those that support privacy and authentication, properties of *usability* as those that improve the client user experience, and properties of *deployability* as those that assist in

the ease of setting up and maintaining a network. Refer to Table 5.1 for definitions on each term being used.

We use this methodical approach to surveying trust models in order to enable the discovery of viable alternatives that can improve upon the properties of the current CA system. We identify the following six trust models for certificate-based authentication in wireless networks:

CA System A heirarchical structure is used where all valid certificates can be traced back to a trusted root certificate via a chain of signatures. The trusted roots are pre-populated on each client device, and every device must be updated whenever a change in root certificates occurs.

Pinning Certificates for networks are pinned (i.e., remembered), with a mapping from the network's SSID. When a network is visited, its certificate will only be accepted if it matches the previously pinned certificate for that SSID. Certificates can either be pinned via a trust-on-first-use (TOFU) model or via an out-of-band distribution method, such as QR codes or NFC tags.

Whitelisting A list of certificates for the most widely-used networks is distributed to the client, with each certificate matching a particular network SSID. Any certificate not found as whitelisted for a particular SSID in question is rejected.

Ad-hoc Local Query The Ad-hoc Local Query parallels multi-path strategies found in website certificate validation, such as the usage of notaries in Perspectives [24]: a technique that samples website certificates from multiple viewpoints in order to overcome local attacks. This model adapts multi-path probing to wireless networks by having clients first sample the certificates provided by local Wi-Fi networks and then share these findings with other co-located clients through an ad-hoc wireless network. Each device would then use these shared findings to attempt to reach a unanimous consensus on the appropriate certificate for a given network, with any variation cuing that a discrepancy might have resulted from a locally-directed evil twin attack.

Collaborative Reporting A central location (website) is used where reputation of network certificates can be crowd-sourced. When connecting to a wireless network, a client queries the site to request the most highly reputed certificate for the SSID of the network in question. This is based on the Collaborative Reporting model proposed by Gonzales et al. [8].

Web of Trust Trust relationships are built in a grassroots fashion, similar to PGP. Rather than using a hierarchical structure based on CAs, a web of trust relies on a distributed system to spread trust locally among smaller groups. Each human end user determines who to trust. This method needs users to be able to share their certificate signatures so other clients can adopt shared certificates into their own store.

5.3 Comparative Analysis

Table 5.2 presents an analysis of the strengths and weaknesses of each trust model. Each row corresponds to a model. Each column stands for a measured property of security, usability, or deployability. For each cell in the table, a ‘ ● ’ represents that the trust model fully satisfies the desired property, a ‘ ○ ’ represents that the trust model conditionally, but not explicitly, satisfies the property, and an empty space represents that the trust model does not satisfy the property. The conditional mark is used for trust models that fulfill an associated property in some cases, but not always. An explanation of each trust model's analysis follows below.

First, we examine the strengths and weaknesses of each model individually. Then, we compare differences between the various models.

Table 5.1: Definition of Properties

Property	**Definition**
First Time Authentication	Able to authenticate a network on the first connection
Successive Authentication	Able to authenticate a network on successive connections
Updatable Certificate	Supports certificate updates/replacements for a network
Supports Revocation	Allows for certificate revocation
Maintains Privacy	No private history information is shared with a third party
No Trusted Third Party	No third party involved in trust relationships
No Reliance on User Feedback	Does not require other users' feedback to validate certificates
SSID as identifier	Supports using SSID as the network identifier
No New User Decisions	Does not introduce any new user decisions
No Internet Connection	No internet connection is required at time of validation
No Additional Hardware Dependency	Nothing beyond standard networking hardware is required
No False Negatives	Does not produce false negatives
No Added Network Expense	No additional monetary cost to the network
Self-signed Certificates	Allows for the use of self-signed certificates
Shared Wi-Fi Networks	Applicable for shared networks
Personal Wi-Fi Networks	Applicable for personal (home) networks
Scalable	Is able to scale to all Wi-Fi networks

Table 5.2: Trust Model Comparative Analysis

		Security							Usability					Deployability				
		First Time Authentication	Successive Authentication	Updatable Certificate	Supports Revocation	Maintains Privacy	No Trusted Third Party	No Reliance on User Feedback	SSID as Identifier	No New User Decisions	No Internet Connection	No Add'l Hardware Dependency	No False Negatives	No Added Network Expense	Self-signed Certificates	Shared Wi-Fi Networks	Personal Wi-Fi Networks	Scalable
Trust Models	CA System	●	●	●	○	●		●			●	●	●			●		●
	Pinning	○	●	○		●	●	●	●		●	●		●	●	●	●	●
	Whitelisting	●	●	○	○	●		●	●	●	●	●	○	●	●	●		
	Ad-hoc Local Query	●	●	●	●			*	●		●		●	●	●	●		●
	Collaborative Reporting	●	●						○			●	○	●	●	●		●
	Web of Trust	○	●	○	○				○		●	●	○	●	●	●	●	
Hybrids	Pinning+CA	○	●	○	○	●	○	●	○		●	●	○		○	●	●	●
	Whitelisting+CA	●	●	○	○	●		●	○	○	●	●	○		○	●		●
	Ad-hoc+CA	●	●	●	○			*	○		●		●		○	●		●
	Coll. Reporting+CA	●	●						○			●	○		○	●		●
	Web of Trust+CA	○	●	○	○	○		○	○		●	●	○		○	●	●	●
	Pinning+Whitelisting	○	●	○	○	●	○	●	●	○	●	●	○	●	●	●	●	●

● = Satisfies, ○ = Satisfies Conditionally

* Requires Real-time Feedback

5.3.1 Individual Evaluation

CA System: Overall, the CA system performs well at providing necessary security guarantees. However, all CAs become trusted third parties and revocation requires a preloaded Certificate Revocation List (CRL). Regarding usability, the CA system does not require any additional hardware and does not produce false negatives, but also does not support the use of SSIDs as an identifier (although this has been initially explored by Byrd et al. [4]) and places the burden on the client to choose which root CA to validate against and whether or not to verify the certificate's identifier. These negatives add complexity to the client's ability to verify a certificate. The CA system also lacks flexibility to support simpler personal network deployments because it places additional costs on the network and does not support self-signed certificates.

Pinning: If certificates are distributed through a TOFU model for pinning, the user has to decide whether to initially trust the network. The corollary to disregarding first time authentication is that the overall system is highly usable and deployable in every environment. A decision must therefore be made about whether it is acceptable to forgo initial authentication on a network to gain these additional benefits. Pinned certificates are also unable to differentiate between an evil twin's certificate and a legitimate certificate update, and thus false negatives are possible on updated certificates or networks with duplicate SSIDs. On the other hand, if an out-of-band distribution channel is utilized for clients to acquire network certificates for pinning, then first time authentication is possible, and updates to certificates can likewise be made through the same channel. Revocation, however, is still lacking since certificates are pinned directly onto client devices without a third party to regulate their validity. Positives of the pinning model include no trusted third parties (because pinning happens locally on the device) and the ability to pin certificates according to an SSID. The potential for running into multiple networks with the same SSID (which is possible due to the lack of a global SSID namespace) is lessened due to certificates being pinned locally.

Pinning also supports self-signed certificates, is scalable, and can be used in all varieties of networks where the security guarantees are acceptable.

Whitelisting: Whitelisting excels overall on security, usability, and deployability characteristics. Its biggest weaknesses are a lack of scalability across wireless networks and the need to ensure the whitelist is updated on each client when a whitelisted certificate has been updated. Thus, false negatives are possible if a whitelist has not been updated before validation. Whitelisting is most effective on high-profile, shared networks where their certificates could be effectively tracked and distributed. It is impractical for personal networks.

Ad-hoc Local Query: The proposed Ad-hoc Local Query model could be used for authentication of all networks but requires real-time feedback (or perhaps cached results) from other clients in order to make an informed decision. Thus, it is unsuitable for personal networks that lack multiple connected devices. Evil twins may also mimic or block feedback from other local devices, which could render this technique ineffective. If no consensus is reached regarding a network's certificate (or if no other client feedback is available), the user is then left to decide whether to trust the network certificate. This model will require an additional network card to gather feedback from others through an ad-hoc network without requiring the other devices to disconnect from their current network connections.

Collaborative Reporting: Collaborative Reporting has little security guarantees beyond security by majority rule. Restraints would have to be placed on user reporting to prevent evil twin attackers from biasing results in their favor. Privacy of network history would not be maintained because the central server would receive a request for each network a client visits and could also track users by their certificate reporting. The system is entirely reliant on user feedback, and it would be hard to allow for certificates updates. Thus, certificate updates could lead to false negative results. Additionally, an Internet connection would be required at the time of connection for any authentication attempt (unless the device was able to anticipate visited networks and cache results.) Thus, the device may require use of a

cellular network in order to fetch the proper network certificate. Clients would also need to decide how to handle cases where a network is not found on the site, and it is unclear how to handle multiple networks with the same SSID.

Web of Trust: The Web of Trust model provides security guarantees by allowing a client to determine which third party signatures to trust and how to build out its own trust relationships. But major questions remain open regarding how to create personal trust relationships and how to distribute signed certificates. If clients decide to trust certificates on the first time connecting to a network, then no first time authentication occurs. Certificate updates and duplicate SSIDs could lead to false negatives, unless the user manually updates their trusted certificate beforehand. Users rely on (and trust) other users in order to build their web of trust. Certificate updates and revocation would also require feedback from other users. The reliance on other users causes this model to be hard to scale. This model also places many decisions directly into the hands of the user (both a potential strength and weakness).

5.3.2 Comparisons to the CA System

Pinning vs. CA: The Pinning model without initial out-of-band certificate distribution lacks the authentication on first use that the CA system provides and does not support automatic revocation and updating, since pinning itself cannot distinguish between a bad certificate and a legitimate replacement. However, Pinning also does not need to rely on third parties, and it flourishes in usability and deployability—a few strengths over the CA system. Additionally, first time authentication for Pinning could be achieved by using out-of-band distribution techniques. Also, unlike the CA system, certificate pinning can be transparent to the user, and an SSID can be used as a Wi-Fi network's identifier. Pinning accepts self-signed certificates, so it supports both shared and personal networks.

Whitelisting vs. CA: As long as the whitelist is kept up to date, the Whitelisting model either provides or builds on every guarantee that the CA system offers, with the exception

that Whitelisting is unable to reasonably scale to all networks. Whitelisting also allows for self-signed certificates and the ability to use a network's SSID as its identifier.

Ad-hoc Local Query vs. CA: The Ad-hoc Local Query (ALQ) model supports both first time and successive authentication, similar to the CA model. However, the ALQ model requires real-time feedback (or caching) in order to make trust decisions and has the clear vulnerability that anyone—including an attacker—is able to pose as a "neutral" party in the ad-hoc network. A reputation system may thus be needed to keep this behavior in check. This model also requires additions to network hardware and presents a new user decision when there is not enough of a consensus to trust a network—two requirements that the CA system does not have.

Collaborative Reporting vs. CA: Collaborative Reporting (CR) is another model that, while supportive of self-signed certificates, lacks many of the security benefits provided by the CA system. It also requires Internet access at the time of validation, while the CA system does not.

Web of Trust vs. CA: The Web of Trust (WoT) is an intriguing alternative as it supports self-signed certificates and allows each client to build out its own trust relationships and maintain its own storage of signed certificates. As a result of having to create individualized trust relationships, however, the WoT requires the client to take responsibility for determining which networks to trust and requires other users' feedback in order to work effectively. The CA system, on the other hand, works autonomously after initial network setup.

5.4 Hybrid Analysis

To both combine strengths and overcome weaknesses of each of these trust models, we also examine whether any benefits might be gained by merging multiple validation strategies into a combined hybrid model.

As a general rule, in Table 5.2 the merging of either 'Satisfies' or 'Satisfies Conditionally' with 'Does Not Satisfy' is represented as 'Satisfies Conditionally.' Exceptions are made on

absolute properties, such as when a cost cannot be overcome through merging or when support of personal networks is maintained through merging of two models.

5.4.1 CA Hybrids

The first four hybrid models shown are extensions to the Certificate Authority model. We target these combinations in order to determine whether any shortcomings of the CA model can be overcome by integration with another validation model.

Pinning+CA: Adding Certificate Pinning, which could be used to accept and pin a network certificate that does not resolve to a trusted root CA, does present the opportunity to expand to SSID identifiers and self-signed certificates for pinned certificates. Still, CA signed certificates do not support these features and the negatives of the CA model concerning cost, usability, and the inclusion of root CAs as trusted third parties remain.

Whitelisting+CA: The combination of Whitelisting with the CA model would allow self-signed certificates to be used for whitelisted networks, while ultimately enabling scalability through the use of the CA model for other non-whitelisted networks. However, it is actually the large distribution of non-whitelisted networks that would benefit most from the reduced cost of self-signed certificates as well as from the use of SSID as the identifier. Neither of these properties are provided by the CA system. Thus, this hybrid still fails to achieve desirable properties of a personal home Wi-Fi network.

ALQ+CA / CR+CA: Ad-hoc Local Query and Collaborative Reporting could each provide a plausible fall-back option for validating certificates that do not resolve to any root Certificate Authorities. However, they are both interesting theoretical models that would ultimately detract from the security performance of the CA model and present many new negatives of their own (e.g., feedback requirements and loss of privacy).

Web of Trust+CA: Adding a Web of Trust to the CA system would allow users to create their own trust relationships without completely relying on CAs. However, negatives

of both models remain, including increased responsibility on end users and the negatives of utilizing CAs.

5.4.2 Whitelisting and Pinning Hybrid

Within the comparative analysis, two models in particular stood out as having complementary strengths and weaknesses: Whitelisting and Pinning. Whitelisting excels at authentication of public networks whose certificates can be effectively tracked and distributed, while Pinning scales across all networks and is highly flexible. Both measured highly on usability and deployability. Therefore, we examined what gains could be made through a combination of these two models.

We found that the hybrid model performs consistently well across every property considered in the analysis. For important networks that require heightened security on first time authentication, the whitelist can be employed, which ensures first time authentication and streamlined updates. For other networks that fall outside of the whitelist, a pinning approach can be used that allows for self-signed certificates, use of SSID as the identifier, and scalability across networks. These other networks could allow TOFU or support a more strategic distribution mechanism, such as the use of QR codes.

A major consideration for this hybrid is how to deliver whitelisted certificates to client devices. One route for delivery could be through operating system channels. This option would follow a push-based strategy, where updates to the whitelist would be pushed out to client devices in the form of an update. This would follow a similar precedent as the current root CA store, which is populated on modern operating systems. An open question here is whether this whitelist would be globally or locally based. Is one master whitelist a viable solution to cover all networks worldwide that should be represented therein, or is there a better solution by using multiple whitelists that differ based on locale? Another route for delivery could instead require clients to actively query a trusted store in order to retrieve a list of network certificates to include in their whitelist. This option would place responsibility

on the client for retrieval, but would also allow flexibility of which origins the client trusts to provide the whitelist.

Chapter 6

Certificate-based Wireless Authentication Platform

We have designed a platform that enables clients to use a variety of trust models to authenticate wireless networks. As shown in Chapter 5, alternative trust models have the potential to provide promising security, usability, and deployability benefits that can further the application of certificate-based authentication in this domain.

6.1 Design

To provide flexibility with different trust models, our system relies on the TrustBase authentication system. TrustBase is a solution developed to intercept and validate server certificates in Internet TLS traffic [18]. TrustBase fixes broken certificate validation within desktop and mobile applications and provides a central hub where alternative certificate validation models can be developed and analyzed. We aimed to build onto TrustBase and leverage its capabilities in order to enable a pluggable interface for certificate validation in wireless network clients. Thus, we designed an external API that enables the Wi-Fi supplicant (as well as any other application) to communicate with TrustBase for certificate validation queries. See Figure 6.1 for a layout of the TrustBase integration. This unification of wireless network validation with web TLS validation enables the research community to collaborate on authentication models to be used in both domains.

We aimed to create a universal architecture that would remain compatible with current wireless network protocols. Therefore, our work modifies only client-side software, enabling the possibility for a step-by-step roll-out that leaves current 802.1X EAP-PEAP

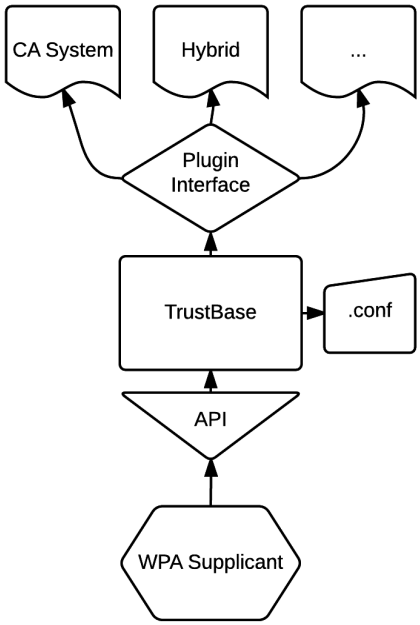

Figure 6.1: TrustBase Integration

network deployments intact and merely requires other networks to upgrade to this previously standardized module in order to enable our solution.

This design also allows users to make individualized choices about how they want to validate a network's certificate. Our modified client permits the user to configure their own methods for validating certificates that are provided through 802.1X's EAP-PEAP authentication module. TrustBase plugins for multiple validation models are available, including plugins for the CA system, Certificate Pinning, and Whitelisting.

6.2 Implementation

We modified the open source *wpa_supplicant* within the Ubuntu Linux distribution in order to extend its certificate validation process during the client's initial 802.1X authentication handshake. Upon receipt of a network's certificate, the supplicant queries TrustBase with a message that includes the SSID and certificate of the network. TrustBase validates the certificate based on the configured plugin that has been set up for validation and returns a

yes/no response to the supplicant. The supplicant then accepts or rejects the TLS handshake with the network, depending on the response it has received.

The wpa_supplicant modification turned out to be a significant and non-trivial step in building this platform. First, we analyzed the open source code to pin-point where the certificate validation occurs within the TLS handshake. We then carefully designed an approach to modifying the existing supplicant that would be as non-obtrusive as possible, while enabling the additional benefits of the TrustBase authentication system. We achieved this by utilizing the existing code structure provided for the OpenSSL TLS handshake, and targeting a specific override of the OpenSSL environment's certificate verification callback function to implement a call to TrustBase's validation engine rather than the typical Certificate Authority root store verification provided by OpenSSL libraries[1].

TrustBase was designed to support interception of public key certificates found within TLS connections at the Application Layer. This interception happens within the operating system kernel and is transparent to all related parties. TrustBase's interception is used as a means of securing broken certificate validation for all applications running on an operating system. However, since the wireless network TLS handshake during 802.1X authentication happens at a lower layer, this communication is not intercepted by TrustBase. Furthermore, we decided that designing a method to intercept wireless network TLS traffic would be inappropriate for our purposes, since we desired to provide a solution integrated directly into the supplicant and did not want to require additional kernel-level access.

Thus, we designed and built an interface that enables both our application as well as other future applications to interact directly with TrustBase's validation engine, instead of needing to rely on its certificate interception. This API was designed to be generalizable and applicable to any application that might want to leverage the capabilities of TrustBase's authentication system directly. The function calls provided by the API allow for the calling application to supply both a certificate chain and a string (such as an SSID) to validate

[1] https://w1.fi/wpa_supplicant/devel/tls__openssl_8c.html

the Common Name of the leaf certificate against. The API call then returns a validation decision based off of the trust model plugins that have been configured within the TrustBase environment.

From our analysis in Section 5.4.2, we identified the Whitelisting and Pinning hybrid model as a highly promising solution for authenticating wireless networks. Accordingly, we built a plugin for TrustBase that implements this hybrid module as a proof-of-concept implementation. The plugin consists of two steps. First, the whitelist is examined to discover whether a certificate has been whitelisted for the current network SSID. If one is found, then the whitelist is enforced: either the incoming certificate matches the whitelisted certificate and is accepted, or else it does not match and is rejected. Alternatively, if a whitelisted certificate for the SSID is not found, then in step two the algorithm uses pinning as a fall-back option. In this second step, the plugin looks for a pinned certificate for the current SSID. If a pinned certificate is found, and the certificate is not expired, the algorithm will accept or reject based on whether the pinned certificate matches the certificate passed in. Otherwise, the certificate in the incoming query will be pinned for that SSID and will be accepted. This plugin uses TOFU for pinning but could also be extended to support other distribution mechanisms for pinned certificates.

6.3 Threat Analysis

In this section, we examine the efficacy of our solution at preventing evil twins in wireless networks. We present an analysis based on the aforementioned evil twin threat model. This includes an assessment of the risks presented by evil twin attacks and an analysis of the effectiveness of our solution at preventing an attack in each of the unique scenarios in which an evil twin can be found.

Evil twins present risks to network clients because an attacker injects themselves as a man-in-the-middle to each client's connection. Without the possibility to perform reliable network authentication, clients are unable to verify whether they are connecting to the proper

network or to a clone that has malicious intent. This attack enables the MitM to compromise the privacy, confidentiality, and integrity of data within the connection stream.

As referenced in the threat model (see Section 4.4), there are four unique variations of an evil twin attack. We first examine the effectiveness of our solution for each variation under the assumption that a valid certificate has been either whitelisted or pinned for the network in question. Afterwards, we discuss the counter-case and possible mitigations.

Replacement Evil Twin In the replacement scenario, an evil twin is placed in the same location as the original network AP, and the original AP is no longer present. Since the certificate-based validation scheme does not depend on location-based tracking, the location of the evil twin is inconsequential. Thus, the validation will still be able to detect that there is a mismatch of certificates in this scenario, even when the location appears to be correct.

Coexistent Evil Twin A coexistent evil twin is co-located with the original AP. Our solution will also be able to differentiate between the evil twin and the original in this scenario, based off of certificate matching. The evil twin's certificate will not match the public key certificate that has been whitelisted or pinned by the client, while the original AP's certificate will match.

Remote Evil Twin A remote evil twin is located at a different location than the original. Once again, since the certificate validation is location agnostic, this scenario is effectively identical to that of the Replacement Evil Twin. The validation will detect that the certificate does not match the original and the TLS handshake will fail.

Ad-hoc Evil Twin An Ad-hoc Evil Twin is one created on the spot in response to probe messages searching for a network by SSID. In this case, even if the attacker does clone the original network's SSID and other identifying information, without the original public/private key pair the evil twin is unable to completely mimic the actual network. The validation will detect that the network certificate does not match.

Despite the ability to detect certificate mismatches in each of the above scenarios, the Whitelisting/Pinning hybrid trust model will fail to detect an evil twin on a first time

connection to a non-whitelisted network in any of these scenarios if a TOFU approach is taken. Likewise, if the certificate for a pinned network is changed, the user must be able to access the updated certificate through an out-of-band distribution, otherwise the user has no way to distinguish between a legitimate certificate update or an evil twin attack. This is one shortcoming that has to be considered when utilizing this trust model. As mentioned previously, a number of strategies could be used to mitigate this drawback, such as an initial out-of-band distribution of a certificate through QR Codes, NFC tags, or a similar technology.

Nevertheless, the ability to validate networks at scale based on their public key certificates would be a monumental step forward in the prevention of evil twin attacks. This protection brings tremendous value in its power to secure sensitive network connections from evil twin attacks.

6.4 Experimental Verification

In our laboratory environment, we built a custom network to experiment with our wireless network authentication platform and verify its effectiveness at defending against evil twin attacks in 802.11 networks. This network consisted of a Linksys WRT54Gv5 wireless broadband router configured to relay authentication requests to a network-internal Windows environment running FreeRadius Version 1.1.7, an open-source RADIUS implementation. The RADIUS server used EAP-PEAP certificate-based 802.1X authentication for all incoming connection requests (including the use of a valid self-signed certificate for network authentication).

We performed multiple simulations on this network to validate that our modified client supplicant was able to utilize both whitelisting and pinning to authenticate a valid network certificate. We then replaced the valid certificate with another certificate in order to simulate an evil twin attack. During these further simulations, we verified that the client was effective at detecting and preventing a connection to the evil twin access point. These tests modeled the client's validation process for each of the four evil twin variations, as previously defined in the threat model.

6.5 Additional Notes

Certificate validation in wireless networking currently depends on the ability to resolve a given certificate to its trusted root authority. This root authority is either a previously installed Certificate Authority that the user trusts by default or a signing certificate provided by the Public Key Infrastructure (PKI) of the enterprise that administers the network. In either case, the client must rely on the hierarchical chain back to the root certificate as sufficient reason to trust the network. The client must therefore trust these root authorities as authoritative sources.

The framework presented in this paper removes the restriction of strict reliance on the CA trust model. This provides increased *trust agility* for the client, as clients are given the option to use alternative validation methods when deciding whether or not to trust a network.

This framework also opens the door to not just using one alternative trust model for all wireless network connections on a device, but also to potentially specifying on a network-by-network basis which validation model to use. For example, it may be applicable for a client to maintain usage of the hierarchical CA model for enterprise networks where a PKI root certificate has been configured, but then switch to a pinning strategy for a home-based network that can be securely pinned on first time use.

Chapter 7

Conclusion

Certificate-based authentication is an effective means of preventing evil twin attacks in 802.11 wireless networks. By applying alternatives to CA-based network certificate validation, we lower the barrier-to-entry of deploying certificate-based network authentication and decouple Wi-Fi certificate validation from a reliance on Certificate Authorities. This, in turn, provides network administrators with an enhanced ability to configure certificate-based authentication, and users will be better protected from evil twins.

The use of TrustBase as a open platform to perform certificate validation on both the Web and wireless spaces empowers the community to research additional trust models for certificate validation. This framework has demonstrated a capability to provide essential characteristics of secure authentication, while decreasing the overhead required for deployment.

Moreover, a thorough analysis of potential alternative trust models for wireless network certificate validation has shown that there are various tradeoffs to be considered among the differing models. We are optimistic that some of these tradeoffs may be offset by further exploring the use of hybrid models, such as a combination of Whitelisting and Pinning certificates. Our analysis suggests that the use of alternative trust models has the ability to raise the overall security, usability, and deployability of certificate-based authentication in wireless networks.

References

[1] Adam Bates, Joe Pletcher, Tyler Nichols, Braden Hollembaek, Dave Tian, Kevin R.B. Butler, and Abdulrahman Alkhelaifi. Securing SSL certificate verification through dynamic linking. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 394–405, New York, NY, USA, 2014. ACM.

[2] R. Beyah and A. Venkataraman. Rogue-access-point detection: Challenges, solutions, and future directions. *IEEE Security and Privacy*, 9(5):56–61, September 2011. ISSN 1540-7993. doi: 10.1109/MSP.2011.75. URL `http://dx.doi.org/10.1109/MSP.2011.75`. Retrieved November 7, 2016.

[3] Sergey Bratus, Cory Cornelius, David Kotz, and Daniel Peebles. Active behavioral fingerprinting of wireless devices. In *Proceedings of the First ACM Conference on Wireless Network Security*, WiSec '08, pages 56–61, New York, NY, USA, 2008. ACM.

[4] Christpher Byrd, Tom Cross, and Takehiro Takahashi. Secure Open Wireless Networking. `https://media.blackhat.com/bh-us-11/Arsenal/BH_US_11_Cross_Arsenal_Secure_Wireless_Slides.pdf`, 2011. Retrieved October 28, 2015.

[5] J. Clark and P.C. van Oorschot. SoK: SSL and HTTPS: revisiting past challenges and evaluating certificate trust model enhancements. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 511–525, Washington, DC, USA, May 2013. IEEE. ISBN 978-0-7695-4977-4. doi: 10.1109/SP.2013.41. URL `http://dx.doi.org/10.1109/SP.2013.41`. Retrieved November 7, 2016.

[6] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. RFC 5280 (Proposed Standard), May 2008. URL `http://www.ietf.org/rfc/rfc5280.txt`. Retrieved November 7, 2016.

[7] P. Funk and S. Blake-Wilson. Extensible authentication protocol tunneled transport layer security authenticated protocol version 0 (EAP-TTLSv0). RFC 5281 (Informational), Aug 2008. URL `http://www.ietf.org/rfc/rfc5281.txt`. Retrieved November 7, 2016.

[8] H. Gonzales, K. Bauer, J. Lindqvist, D. McCoy, and D. Sicker. Practical defenses for evil twin attacks in 802.11. In *Proceedings of the IEEE Global Telecommunications Conference*, GLOBECOM '10, pages 1–6, Miami, Florida, USA, 2010. IEEE.

[9] Hao Han, Bo Sheng, C.C. Tan, Qun Li, and Sanglu Lu. A timing-based scheme for rogue AP detection. *IEEE Transactions on Parallel and Distributed Systems*, 22(11): 1912–1925, Nov 2011.

[10] Help Net Security. RSA 2007: More than 150 wireless devices onshow floor at RSA Conference vulnerable to attacks. `http://www.net-security.org/secworld.php?id=4749`, Feb 2007. Retrieved October 31, 2015.

[11] P. Hoffman and J. Schlyter. The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA. RFC 6698 (Proposed Standard), Aug 2012. URL `http://www.ietf.org/rfc/rfc6698.txt`. Retrieved November 7, 2016.

[12] Fabian Lanze, Andriy Panchenko, Benjamin Braatz, and Thomas Engel. Letting the puss in boots sweat: Detecting fake access points using dependency of clock skews on temperature. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '14, pages 3–14, New York, NY, USA, 2014. ACM.

[13] Fabian Lanze, Andriy Panchenko, Ignacio Ponce-Alcaide, and Thomas Engel. Undesired relatives: Protection mechanisms against the evil twin attack in IEEE 802.11. In *Proceedings of the 10th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '14, pages 87–94, New York, New York, USA, Sept 2014. ACM.

[14] Fabian Lanze, Andriy Panchenko, Ignacio Ponce-Alcaide, and Thomas Engel. Hacker's toolbox: Detecting software-based 802.11 evil twin access points. In *Proceedings of the 2015 12th Annual IEEE Consumer Communications and Networking Conference*, CCNC '15, pages 225–232, Las Vegas, Nevada, USA, 2015. IEEE.

[15] Frederic Lardinois. Study: 61% of U.S. households now have WiFi. `http://techcrunch.com/2012/04/05/study-61-of-u-s-households-now-have-wifi/`, Apr 2012. Retrieved October 31, 2015.

[16] B. Laurie, A. Langley, and E. Kasper. Certificate transparency. RFC 6962 (Experimental), Jun 2013. URL `http://www.ietf.org/rfc/rfc6962.txt`. Retrieved November 7, 2016.

[17] Moxie Marlinspike. SSL and the Future of Authenticity. http://www.rsaconference.com/writable/presentations/file_upload/ht2-107.final.pdf, 2012. Retrieved November 9, 2015.

[18] Mark O'Neill, Scott Heidbrink, Jordan Whitehead, Scott Ruoti, Dan Bunker, Kent Seamons, and Daniel Zappala. Trustbase: An architecture to repair and strengthen certificate-based authentication. *arXiv:1610.08570*, 2016.

[19] R. Oppliger. Certification authorities under attack: A plea for certificate legitimation. *IEEE Internet Computing*, 18(1):40–47, Jan 2014.

[20] George Ou. A secure wireless LAN hotspot for anonymous users. http://i.i.cbsi.com/cnwk.1d/i/tr/downloads/home/anonymous_hotspot.pdf, Jul 2007. Retrieved October 28, 2015.

[21] D. Simon, B. Aboba, and R. Hurst. The EAP-TLS authentication protocol. RFC 5216 (Proposed Standard), Mar 2008. URL http://www.ietf.org/rfc/rfc5216.txt. Retrieved November 7, 2016.

[22] Christopher Soghoian and Sid Stamm. Certified lies: Detecting and defeating government interception attacks against SSL (short paper). In *Proceedings of the 15th International Conference on Financial Cryptography and Data Security*, FC '11, pages 250–259, Berlin, Heidelberg, 2012. Springer-Verlag.

[23] Xinli Wang, Yan Bai, and Lihui Hu. Certification with multiple signatures. In *Proceedings of the 4th Annual ACM Conference on Research in Information Technology*, RIIT '15, pages 13–18, New York, NY, USA, 2015. ACM.

[24] Dan Wendlandt, David G. Andersen, and Adrian Perrig. Perspectives: Improving SSH-style host authentication with multi-path probing. In *Proceedings of the USENIX 2008 Annual Technical Conference*, ATC '08, pages 321–334, Berkeley, CA, USA, 2008. USENIX Association.

[25] Wi-Fi Alliance. Wi-Fi alliance "fifteen for 2015" predictions. http://www.wi-fi.org/beacon/wi-fi-alliance/wi-fi-alliance-fifteen-for-2015-predictions, Jan 2015. Retrieved October 31, 2015.

[26] Glen Zorn, Ashwin Palekar, Daniel Simon, and Simon Josefsson. Protected EAP protocol (PEAP). https://tools.ietf.org/html/draft-josefsson-pppext-eap-tls-eap-06, 2003. Retrieved November 7, 2016.