Mitigating Evil Twin Attacks in 802.11

Kevin Bauer, Harold Gonzales, and Damon McCoy
Department of Computer Science
University of Colorado
{bauerk, gonzaleh, mccoyd}@colorado.edu

Abstract— **Due to the prevalence of insecure open 802.11 access points, it is currently easy for a malicious party to launch a variety of attacks such as eavesdropping and data injection. In this paper, we consider a particular threat called the *evil twin attack*, which occurs when an adversary clones an open access point and exploits common automatic access point selection techniques to trick a wireless client into associating with the malicious access point. We propose two lines of defense against this attack. First, we present an evil twin detection strategy called *context-leashing* based upon recording the nearby access points when first associating with an access point. Using this contextual information, the client determines if an adversary has setup an evil twin access point at a different location. Next, we propose an SSH-style authentication method called *EAP-SWAT* to perform one-way access point authentication that fits into the extensible authentication protocol (EAP) framework.**

I. Introduction

According to a recent study, 42% of wireless 802.11 access points (APs) provide no security mechanisms — not even WEP or WPA [1]. Often times, wireless APs are left open for convenience. For example, a coffee shop or bookstore may wish to offer a free wireless service, so there is no need to authenticate its wireless users. However, wireless clients that use these APs are vulnerable to a number of trivial threats such as eavesdropping and injection attacks. An additional and often over-looked vulnerability caused by using open APs is the access point impersonation attack. This is commonly referred to as the *evil twin attack* and occurs when a client is tricked into associating to a malicious rogue AP with the same identity (or SSID) as a previously-used open AP [2]. An adversary can use an evil twin as a platform to launch a variety of attacks. For instance, the evil twin could hijack the DNS mechanism and redirect the clients to malicious servers that launch phishing attacks or attempt to install malware.

To make matters worse, the client is vulnerable to an evil twin attack even when communicating with an AP that deploys traditional security mechanisms. For example, once a user has chosen to connect to a particular network, it is added to the user's *preferred network list*, which is provided by most modern operating systems. When a client is searching for a network, it will send probe messages for the networks on its preferred network list. Then, an attacker intercepts these probe messages and deploys an evil twin presenting the same network name as one of the client's preferred networks. When the client sees a network from its preferred network list, it will automatically and transparently associate to this AP, which can be an evil twin. Even if secure networks are available, the attacker can prevent the client from using a secure access point or disrupt an existing connection by launching one of several denial-of-service attacks [3] and force the client to associate to the evil twin. At this point, the evil twin can launch one of several attacks without the client's knowledge, or any perceived loss of security.

The ease of deploying an evil twin AP highlights the inherent difficulty of establishing identity that exists in wireless networks. Without any type of authentication mechanism, a client ultimately does not know the identity of their AP. The current 802.11 authentication mechanisms within WEP or WPA to establish identity require pre-shared secrets that must be communicated out-of-band. However, this is inconvenient for a coffee shop network where the clients may not be known in advance. Our goal is to provide an easy mechanism to establish an AP's identity that is convenient for both the clients and APs and requires no pre-shared secrets.

We present two strategies to defend against evil twin attacks by establishing an AP's identity. The first approach, called *context-leashing*, records contextual location information which consists of the other access points that are visible to a wireless client from the location when first connecting to an access point. This limits the location in which the AP's identity can be trusted to the one the client recorded when first associating to an AP. If an adversary attempts to clone this AP at another location, the context of visible APs will be different, and thus, the client should use additional caution or reject this potentially malicious AP. This approach requires no cooperation from the APs and can be deployed at will by individual clients. While this approach has been previously applied to enable device localization [4], to the best of our knowledge we are the first to use this technique to detect evil twin APs. This defense protects clients from an evil twin in the wrong context. However, clients are still vulnerable to this attack when the adversary clones an AP within the correct context. To defend against the attack within the correct context, it is necessary to provide a stronger form of identity for APs.

We propose that access point identities be bound to cryptographic credentials. However, to avoid the inconvenience of sharing keys out-of-band, APs should share public keys in the style of SSH. To this end, we propose a new authentication method for the 802.1X extensible authentication protocol (EAP) called the Simple Wireless Authentication Technique, or *EAP-SWAT*. Like SSH, this method follows the principle of *trust-on-first-use* and allows the client to establish a shared secret session key with a desired AP. While this approach requires participation from the AP, it provides a stronger form of AP identity than can be achieved using context-leashing. One limitation of the trust-on-first-use approach is that it is ostensibly vulnerable to man-in-the-middle attacks the first time the client contacts the service. However, this provides a considerable improvement over open access points because clients are no longer vulnerable to the evil twin attack after they associate with the AP for the first time.

978-1-4244-3367-4/08/$25.00 ©2008 IEEE

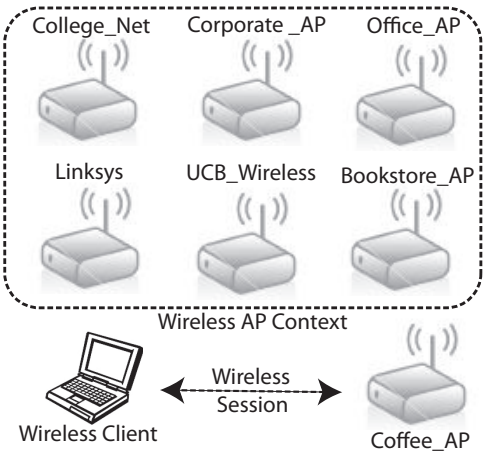

Fig. 1. The wireless access point context for a sample wireless session

II. Context-Leashing Evil Twin Mitigation

We first propose a technique to detect evil twin access points that can be deployed without any modifications to the access points using only contextual information. Here, contextual information refers to the list of all access points that are visible to a client when it is associated with a particular wireless network at a familiar location. For example, suppose that a client associates to an access point with an SSID `Coffee_AP` at their favorite coffee shop. While associated, the client can hear beacon messages from several other access points, indicating that there are several other wireless networks nearby (as shown in Figure 1). These other nearby SSIDs define the context for the `Coffee_AP` SSID. Now suppose that the client observes a wireless network with SSID `Coffee_AP` at another location such as an airport where the context is significantly different. In this case, the client should exercise caution about associating to this access point as it is likely to be an evil twin. For this technique, we assume that all APs are non-mobile.

More formally, let $C_i = \{c_1, ..., c_j, ..., c_n\}$ be the set of all SSIDs that are visible from location i when associated with an access point with SSID c_j. The context for SSID c_j is $C_i \backslash \{c_j\}$. When a client associates to an SSID for the first time, the context will be unfamiliar. In this case, the client *learns* the context for this new SSID. The client maintains a set of learned contexts for each SSID to which it associates. When a client sees an SSID again, it must compare the observed context with the previously learned context for that access point.

To match an observed context o with one of the learned contexts, it is necessary to apply a metric to capture set similarity. In Equation 1, the expected context $e_i \in E$ that maximizes the Jaccard similarity coefficient is found:

$$J = \underset{e_i \in E}{\operatorname{argmax}} \frac{|o \cap e_i|}{|o \cup e_i|} \quad (1)$$

To determine if an observed context is sufficiently close to the expected context for a particular SSID, a threshold τ can be derived. If $J > \tau$, then the context is within an acceptable proximity and the access point is accepted. Otherwise, the access point is regarded as a potential evil twin. In this case, the client should not transparently associate to this AP. If the client does, in fact, still wish to associate despite the danger, then the user may manually associate.

A. Evil Twin Detection Example Using Context

Suppose that the wireless client from Figure 1 leaves the coffee shop and moves to another location. The wireless client immediately probes for its preferred networks, including `Coffee_AP`. A malicious access point hears the probe request and creates an evil twin open AP called `Coffee_AP` and the client would typically automatically and transparently associate to this malicious AP.

Now, suppose that the client uses contextual information to verify that this AP is in the expected location. If there are six other APs at this new location that have not been seen before, then the Jaccard similarity coefficient is $\frac{1}{12}$ which is sufficiently small to reject this access point.

B. Discussion

This context-leashing technique can detect evil twin access points while requiring no modifications to the access points or any of the wireless infrastructure. This property makes this approach very easy to deploy. In addition, this detection method makes it more difficult for a malicious access point to lure an unsuspecting client from a secure wireless network to an insecure one with a stronger signal that is in the client's preferred network list. In addition, this technique works even when there are no other APs in a particular AP's context. In this case, an evil twin may be detected if it is setup in another location with several APs.

However, this approach does not provide any form of authentication and does not provide any confidentiality mechanism to prevent against injection and eavesdropping. In fact, regardless of whether evil twin access points are detected and ignored, any arbitrary wireless device can perform injection and eavesdropping attacks on an open wireless network. Therefore, it is essential to provide secure data delivery in wireless networks. In the next section, we present a simple protocol to provide access point authentication and session key establishment to enable secure data delivery.

III. *EAP-SWAT* Design

In this section, we present the design of a simple access point authentication method that fits into the EAP framework. *EAP-SWAT* provides a one-way authenticated TLS session where the client authenticates the AP. The sequence of messages in *EAP-SWAT* is essentially the same as EAP-TTLS [5] in the one-way server authentication mode. However, we define a distinct authentication type to notify the client that they should obey the principle of trust-on-first-use with the AP's certificate.

A. The Extensible Authentication Protocol

The extensible authentication protocol (EAP) is an authentication framework where the method of authentication is defined in a pluggable module that can be chosen to suit the authentication task at hand [6]. EAP is a request-response protocol and proceeds as follows: First, a *request* message is sent from the authenticator containing the specific type

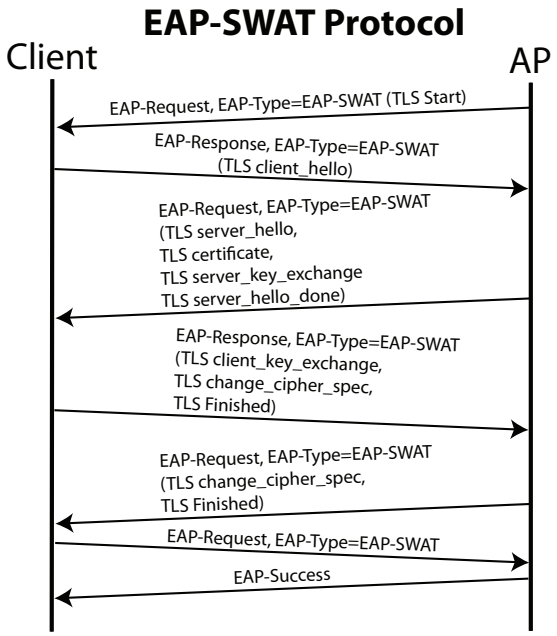

Fig. 2. The sequence of messages for EAP-SWAT

of authentication that is to be performed. For instance, the authentication type refers one of the authentication methods such as MD5-challenge or challenge-response (CHAP). Next, the other party sends a *response* message agreeing to the authentication method. A series of request-reply messages proceed during which the actual authentication is conducted. At the end, either a success or failure message is sent by the authenticator.

B. Certificate Identity Binding

Certificates are commonly used to bind public keys to a particular identity. For instance, in TLS public keys are bound to a server's identity using a URI such as a DNS name. Certificate authorities (CAs) such as VeriSign establish a root of trust and can be used to verify the authenticity of a server's public key certificate. To establish a one-way authenticated TLS session with an access point, it is necessary to bind the AP's public key to its identity. However, APs have no strong identities such a URI, so it's difficult to use a CA to verify an AP's certificate. Therefore, APs provide self-signed certificates. The AP's identity can be expressed as either its SSID or its unique MAC address. The SSID may be a useful identifier in the case of a large multi-access point corporate or university network where transparent hand-off between APs is desired. Alternatively, binding certificates to a MAC address may be beneficial when the SSID is common or there is only a single access point for an SSID (such as a home or coffee shop network). To make the MAC address binding more human-readable, one could combine the MAC/SSID pair into a single identifier.

C. Protocol Overview

The sequence of messages involved in the *EAP-SWAT* authentication method is shown in Figure 2. After the client sends an 802.11 authentication request message, the AP initiates the 802.1X protocol [7]. 802.1X initiates EAP to perform the authentication, which proceeds as follows: First, the AP sends an EAP-Request to establish the authentication method, which is EAP-SWAT. The client acknowledges the selection of this authentication type with an EAP-Response message and initiates a TLS session with a TLS client_hello message. Next, the AP replies with a TLS server_hello, its public key certificate, a desired method for session key exchange, and the TLS server_hello_done. The client sends a message to acknowledge the key exchange method and chooses the cryptographic parameters. The AP acknowledges the cryptographic parameters and the client and AP exchange a final pair of messages to determine if the authentication is successful. After the authentication phase succeeds, the client and AP can agree upon a session key using RSA. This protocol differs from EAP-TTLS [5] since the client must decide whether or not to accept the server's certificate, while EAP-TTLS requires that credentials be exchanged out-of-band before authentication takes place.

The one-way authenticated TLS session provides protection against replay attacks and ensures forward secrecy, which means that if a session key is compromised, then only the current session is compromised (*i.e.*, encrypted data captured from previous sessions remains secure).

D. Key Updating

It is common for a certificate to have an expiration date such as one year after its issue. To issue a new public key, the AP signs the newly generated public key with its old private key (before the old key pair expires). When the client is presented with a new key for a familiar AP, it simply verifies the signature and accepts the new public key only if the signature is correct. However, if the client does not receive an updated public key, then the client must remove the old expired public key and perform another trust-on-first-use authentication to acquire the new key.

E. SSID Conflict Resolution

Suppose that a client sees the same SSID in different locations. For instance, the client connects to an AP with SSID `linksys` at home and also desires to connect another access point with the same SSID at the coffee shop. There are three possible ways to deal with this identity conflict.

First, the client could force the AP to provide a certificate with an identity based on a MAC/SSID pair. This resolves the ambiguity, but now makes it difficult for the client to transparently migrate from one AP to another within the same SSID as is common in corporate or university wireless networks.

Second, the client could remove the conflicting certificate from its local certificate chain. In the case, the client would use the certificate for SSID `linksys` 1, but then when they returned to the `linksys` 2 network, they would need to perform the one-way AP authentication procedure again. This type of thrashing behavior would result in additional overhead to perform the authentication many additional times unnecessarily.

Third, the client could record each network's context and use the contextual information to resolve SSID ambiguities. Using the context, the client can select correct certificate for the desired SSID.

F. Discussion

EAP-SWAT provides an automated one-way authentication mechanism that does not require any out-of-band key exchange. However, since this authentication approach is based on the principle of trust-on-first-use, then it is vulnerable during the first authentication. We argue that this is not a fatal limitation, since the first time a client associates with an access point is not automated and requires the user to manually select the specific access point they wish to use. Also, the user is aware of their location and can use this information to select a more trustworthy access point. Subsequent automatic associations to the access point are thereafter authenticated and secure.

While the protocol as previously described performs server authentication only, it can be extended to authenticate the client also. In this case, the client would provide a certificate with a client ID or user name to bind to the client's certificate. The AP can now use this authentication to monitor the client's activity, for instance, to respond to reports of network abuse. This would be an improvement over using MAC addresses to grant access to the network and monitor a client's usage.

IV. Possible Attacks

While the techniques proposed provide defenses against access point selection attacks such as the evil twin attack, it is possible that these defenses themselves could be exploited. One straight-forward attack against the context-leashing evil twin detection method would be for an adversary to record the context for a particular target AP that they wish to clone and then also clone each AP to replicate the context. This attack requires that the adversary know a large portion of the context for the target AP, which may not always be possible. Also, it is only possible to re-create a target AP's context in an environment that has no 802.11 APs — otherwise, the context would be different. We argue that effectively replicating a particular AP's context is sufficiently difficult for a modest adversary. However, recall that this technique does not provide a secure session (*i.e.*, with authentication, integrity, and confidentiality) and consequently, insecure sessions are *always* vulnerable to eavesdropping and data injection attacks.

To address the limitations of the context-aware approach, we present an authentication mechanism for EAP to perform one-way AP authentication. However, this approach is vulnerable the *first time* that the AP is used. As in other trust-on-first-use protocols such as SSH, the client must decide to accept the public key for the given AP.

In addition, it is possible for an adversary to impersonate a target AP and present clients with a fraudulent public key. However, it is difficult to distinguish between an AP revoking a potentially compromised key and an evil twin attack. The client would need to decide to trust the new key or reject it, since there is no certification authority in our proposed architecture to vouch for the new key. We can use contextual information to detect the case of a malicious AP presenting a fraudulent key as follows. If a new key is presented in a different context than expected by the client, then the new key and AP should be rejected. Similarly, if a malicious AP attempts to clone another AP within the expected context, then the client would detect two APs with the same identity, but with differing keys. In this case, the client should detect the original AP with the correct key and simply reject the impersonator. Therefore, it is important to use the context-leashing in conjunction with *EAP-SWAT* to mitigate this attack.

V. Conclusion

We presented two simple defense strategies to mitigate the evil twin attack in 802.11 access point selection. The first method uses contextual information to allow clients to learn the context of other access points around which a particular AP should be trusted. However, this approach does not provide security for a client's session in the form of authentication, integrity, and confidentiality, so the client is still vulnerable to a variety of other attacks such as eavesdropping and data injection. To address this limitation, we propose *EAP-SWAT*, a simple authentication method that fits into the extensible authentication protocol (EAP) framework under 802.1X. This approach — based upon the concept of *trust-on-first-use* — provides one-way AP authentication and a mechanism to establish a shared secret key to create a secure session.

Our approach requires no additional infrastructure and only minimal modifications to the access points to support *EAP-SWAT*. In addition, the proposed strategy requires no pre-shared secrets or out-of-band communication between the clients and APs; thus, it offers convenience and is highly practical. However, the trust-on-first-use approach has limitations — namely, that the client may be vulnerable the first time that the AP is used. But given the prevalence of open APs, these defenses lessen the threat of evil twin APs.

VI. Acknowledgements

We would like to thank Shivakant Mishra and the anonymous reviewers for their helpful comments.

References

[1] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved access point selection," in *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, 2006.

[2] D. Dai Zovi and S. Macaulay, "Attacking automatic wireless network selection," *Information Assurance Workshop. IAW '05: Proceedings from the Sixth Annual IEEE SMC*, June 2005.

[3] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *SSYM'03: Proceedings of the 12th USENIX Security Symposium*, 2003.

[4] A. Lamarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place Lab: Device positioning using radio beacons in the wild," in *Pervasive*, 2005.

[5] P. Funk and S. Blake-Wilson, "RFC 5281: Extensible Authentication Protocol Tunneled Transport Layer Security," Network Working Group, August 2008.

[6] B. Aboba *et al.*, "RFC 3748: Extensible Authentication Protocol (EAP)," Network Working Group, June 2004.

[7] "802.1X Standard," IEEE, July 2004.