Evil Twins and WPA2 Enterprise: A Coming Security Disaster?

Alberto Bartoli[1], Eric Medvet, Filippo Onesti —— University of Trieste, Italy

Abstract

WPA2 Enterprise is a suite of protocols for secure communication in a wireless local network and has become an essential component of virtually every enterprise. In many practical deployments of this technology, a device that authenticates with username and password is at risk of leaking credentials to fraudulent access points claiming to be the enterprise network (*evil twins*) that may be placed virtually anywhere. While this kind of vulnerability is well known to practitioners, we believe these issues deserve a fresh look because the current technological landscape has magnified the corresponding risks. Convergence of organizations toward single sign-on architectures in which a single set of credentials unlock access to *all* services of the organizations, coupled with the huge diffusion of wifi-enabled personal devices which often contain enterprise credentials and that connect to wifi networks *automatically*, have made attacks aimed at stealing network credentials particularly attractive to attackers and hard to detect. In this paper we intend to draw the attention of the research and technological community on this important yet, in our opinion, widely underestimated risk. We also suggest a direction for investigating practical solutions able to offer stronger security without requiring any overhaul of existing protocols.

Keywords: authentication, wifi, smartphone, hacking, password

1 Introduction

WPA2 Enterprise is a suite of protocols for secure communication in a wireless local network and has become an essential component of virtually every enterprise. The framework is based on three different entities: user device that communicates via wireless link (Supplicant); wireless access point (Authenticator); server that stores user credentials (Authentication Server). Each user is provided with personalized credentials for authenticating to the network and authentication may occur with a variety of protocols. Successful connection ensures secrecy, integrity and mutual authentication of the wireless traffic between Supplicant and Authenticator.

While the message flow depends on the specific combination of authentication protocols used, the overall exchange follows a common pattern. The Authentication Server sends the Supplicant a certificate binding the name of the server to a public key. Then, the Supplicant proves knowledge of the user credentials with a message exchange occurring within a TLS tunnel built upon the public key in the received certificate. We emphasize that the certificate does not bind a public key to the name of the wireless network (i.e., to its SSID), only to the DNS name of the Authentication Server.

[1] Corresponding author: http://bartoli.inginf.units.it

An out of the box Supplicant cannot connect securely to a given SSID with WPA2 Enterprise, unless the user knows the DNS name of the Authentication Server and checks that the certificate sent in the early stage of the authentication protocol is signed by a trusted CA and contains that name. Indeed, there are a number of simple means for a rogue access point to claim to be the target SSID and provide a certificate that will be validated by the Supplicant automatically (e.g., [Y13,SH13,M13]). For this reason, WPA2 Enterprise Supplicants must be properly configured *before* connecting to the target SSID [SS12]. The nature of such configuration depends on the authentication protocols available and its objective consists, essentially, in binding the SSID to the certificate of the Authentication Server, making it impossible for the Supplicant to accept other certificates for that SSID.

The requirement for secure configuration of a Supplicant before connecting to a WPA2 Enterprise network is well known and the attacks that can be carried out when such a requirement is not fulfilled are well known as well (an excellent synthesis can be found, e.g., in [WWW15]). However, we believe that these issues deserve a fresh look because the current technological landscape has magnified the risks associated with *evil twins*, i.e., with rogue access points that claim to be associated with an enterprise SSID.

First, convergence of modern organizations toward single sign on (SSO) architectures in which credentials for wireless network authentication usually unlock access to *all* enterprise services, makes attacks aimed at stealing network credentials particularly attractive. Second, the fact that virtually every enterprise user is now permanently carrying a personal wifi-enabled smartphone which often contains the user's enterprise credentials and connect to wifi networks *automatically*, makes those attacks very simple to execute and very difficult to detect. The reason is because an evil twin (ET) may trick a device into initiating an authentication protocol execution easily and, as we shall illustrate in more detail later, even a failed protocol execution often suffices to leak credential material to the attacker. The crucial observation is that attacks of this kind may be executed automatically, in less than a second of proximity to an ET, and without any need of involving the device owner in a working session. It follows that these attacks may occur even outside of the enterprise and potentially anywhere: the ET may quickly elicit credential material from the device and then disconnect immediately. Furthermore, the ET does not even need any Internet connectivity. Preventing client-owned devices of enterprise users to fall prey of these attacks requires a mix of careful user education, correct user behavior and suitable configuration of user devices: a combination very hard to obtain in practice. The resulting scenario could make WPA2 Enterprise prone to a widespread security disaster soon.

In this paper we intend to draw the attention of the research and technological community on this important yet, in our opinion, widely underestimated risk. We also suggest a direction for investigating practical solutions able to offer stronger security without requiring any overhaul of existing protocols.

An important class of organizations included in our analysis are the research and educational institutions belonging to the eduroam network, which is a roaming access service allowing members of participating institutions to obtain Internet connectivity at any other participating institution. In 2016, eduroam provided over 2.6 billion authentications of roaming users in the same country and over 592 million cross-border authentications (https://www.eduroam.org/2017/03/07/2016-a-record-breaking-year-for-eduroam/). Eduroam connectivity is increasingly available even outside of campuses of participating institutions through hot-spots at city centers, commercial malls, airports and railway stations and so on: tens of thousands of locations in more than 85 countries world-wide (https://www.eduroam.org/case-studies/). Essentially, eduroam provides a secure infrastructure for enabling Supplicants to execute the WPA2 Enterprise authentication protocol with the

Authentication Server of their home institution rather than with the Authentication Server of the institution they happen to be [MW12, WWW15]. To this end, Authentication Servers of participating institutions route authentication messages within the TLS tunnel established between a Supplicant and its home Authentication Server.

It is useful to place our attack scenario in perspective with respect to the recently discovered KRACK vulnerability in WPA2 [VP17]. KRACK allows an attacker in range of a WPA2 network to perform arbitrary packet decryption and, in most network configurations, arbitrary packet injection. The impact of KRACK depends on the protocols used for the wifi payload and may include loss of confidentiality as well as TCP connection hijacking. KRACK does *not* allow the attacker to obtain WPA2 credentials, nor does it allow the attacker to obtain wifi encryption keys. KRACK exploits weaknesses of the WPA2 standard, thus any correct implementation of the standard was likely affected at the time KRACK was disclosed[2]. In this work we are concerned with a very different scenario. First, we consider attacks that do not occur while in range of an enterprise network: an ET may placed anywhere. Neither the attacker nor the ET need to have ever been in range of the enterprise network. Second, we consider attacks aimed at stealing WPA2 credentials. Finally, we consider attacks that may only affect Supplicants that are not configured correctly. Our scenario is thus complementary to the one relevant to KRACK.

2 Supplicant configuration

Supplicants, i.e., user devices that connect to a WPA2 Enterprise network, must be configured with the *network profile* for the target SSID, that is, they must be provided with a set of rules enforced by the networking software for binding the SSID of the network to the Authentication Server of that network.

In principle, only Supplicants configured with the correct network profile should be used in a WPA2 Enterprise environment, because Supplicants that are not configured correctly are prone to several security vulnerabilities [SS12,WWW15]. For example, an ET could impersonate the target SSID and provide a certificate that is validated automatically by the Supplicant because certificates bind a public key to a server name, rather than to a SSID. In that case a Supplicant that contains networking credentials and connects to the target SSID automatically, will send credential material to the ET without any involvement of the user (WPA2 Enterprise authentication will be described in more detail in the next sections, along with the corresponding attacks).

In practice, ensuring that Supplicants indeed install the correct network profile may be costly and difficult, especially in organizations with hundreds or thousands of users such as, e.g., universities. Indeed, according to a recent survey among 880 IT professionals of organizations allowing enterprise usage of personal devices (Bring Your Own Device, BYOD), 23% of organizations offered no technical support and leave users responsible for configuring and supporting their personal devices; 27% offer some limited support and 32% offer a best-effort support without any formal process or capabilities [C16]. Although BYOD support is a much broader theme than mere connectivity to the enterprise network, the overall issue is evident.

A crucial problem is that Supplicants may connect to an enterprise wireless network even without installing the correct network profile. What often happens is that users insert their enterprise credentials in the personal devices, select the SSID of the enterprise network and then play with the network configuration until connecting, perhaps instructing their devices to skip certificate validation (an option available in almost all operating systems). According to a recent survey among students at a research University in Russia, more than 40% of participants adopt exactly

[2] Further information may be found also at https://www.krackattacks.com/ and https://www.kb.cert.org/vuls/id/228519/

this behavior [Y16]. Anecdotal evidence leads us to believe that for many institutions, including ours, the percentage of Supplicants which skip certificate validation is much higher.

Discouraging unsafe practices by allowing connections only from Supplicants that are configured with the correct network profile is usually not possible: successful authentication of a Supplicant with the legitimate authentication server typically does not allow the latter to infer whether the network profile of the former is configured correctly. It follows that the security properties of the network connection critically depend on the collaboration of end users, which should be educated to not even *attempt* a connection before configuring their devices properly. Most importantly, these security properties cannot be enforced automatically for misbehaving users.

Adopting a security policy in the enterprise that forbids such insecure practices is important but cannot be the only solution, because users can hardly appreciate the risks involved in connecting without installing the correct network profile. In this respect, we may also observe that wireless connection at home does not require any specific configuration on client devices other than password insertion; and, that the need of providing personalized credentials does enforce a feeling of security. Indeed, according to a survey by Microsoft, 24% of respondents use their personal devices in the workplace even in organizations that explicitly ban the use of those devices, while a further 23% use their personal devices in the absence of any specific policy or without even knowing whether such a policy exists [J12].

Many organizations are not particularly restrictive concerning usage of personal devices, though. According to a recent survey in 500 US companies with more than 1000 employees, half of the organizations that support or mandate usage of personal devices do not have a formal policy for their usage [BL16]. Another recent survey in companies with more than 100 employees led to similar results [S16]. It is also important to remark that we observed user guides for eduroam connectivity at several institutions which do not require any automatic installation of an approved network profile and instruct users to select such options as “do not validate the certificate”. Such choices do not violate the eduroam policy because user devices are only required to support at least one of the EAP protocols capable of mutual authentication [MW12].

In summary, although personal devices should be properly configured before connecting to the enterprise wireless network, very often this is not the case in practice. Rather than blaming users and/or organizations and dismissing this fact as a bad practice that should be avoided, we believe that this issue should be considered as a reality and that we should understand its implications.

2.1 WPA2 Enterprise authentication

WPA2 is a large suite of protocols. We provide only the necessary background and focus on the configurations most commonly used for authentication in enterprise wireless networks. We refer the reader to the abundant literature on the subject for full details, e.g., [FEOS07]. WPA2 Enterprise authentication protocols are EAP-TLS, EAP-TTLS and PEAP. In EAP-TLS user credentials consist of a private-public key pair and a certificate binding the user identity to that public key. If an organization can afford to deploy EAP-TLS and enforce its usage across all client devices, then the issues discussed in this work are not relevant. The reason is because, with EAP-TLS, the Supplicant proves knowledge of a private key and does not send any password or token, thus tricking the Supplicant into executing the authentication protocol with an ET cannot leak any credential information to the attacker. On the other hand, deploying EAP-TLS in a large organization is difficult and costly due to the problems involved in key and certificate management for Supplicants. In the following we shall consider only organizations based on either EAP-TTLS or PEAP and such that user credentials consist of a pair username-password. We are not aware of any statistics quantifying the portion of WPA2 Enterprise deployments that fall in this category, but it is fair to claim that such a setting is indeed very common.

With both EAP-TTLS and PEAP, execution of the authentication protocol occurs in two phases, as follows (see also Figure 1). In Phase 1, the Supplicant declares the user identity for the advertised SSID and the Authentication Server responds with a certificate binding the name of the server to a public key. Then, in Phase 2, the Supplicant proves knowledge of the user credentials with a message exchange occurring within a TLS tunnel built upon the public key in the received certificate. Both EAP-TTLS and PEAP are compound protocols able to transport a variety of different authentication protocols within the TLS tunnel, e.g., MSCHAPv2. Depending on the specific protocol used in Phase 2, the Supplicant may send either the password or a hash of the password; and, the Authentication Server may or may not prove knowledge of the user credentials to the Supplicant.

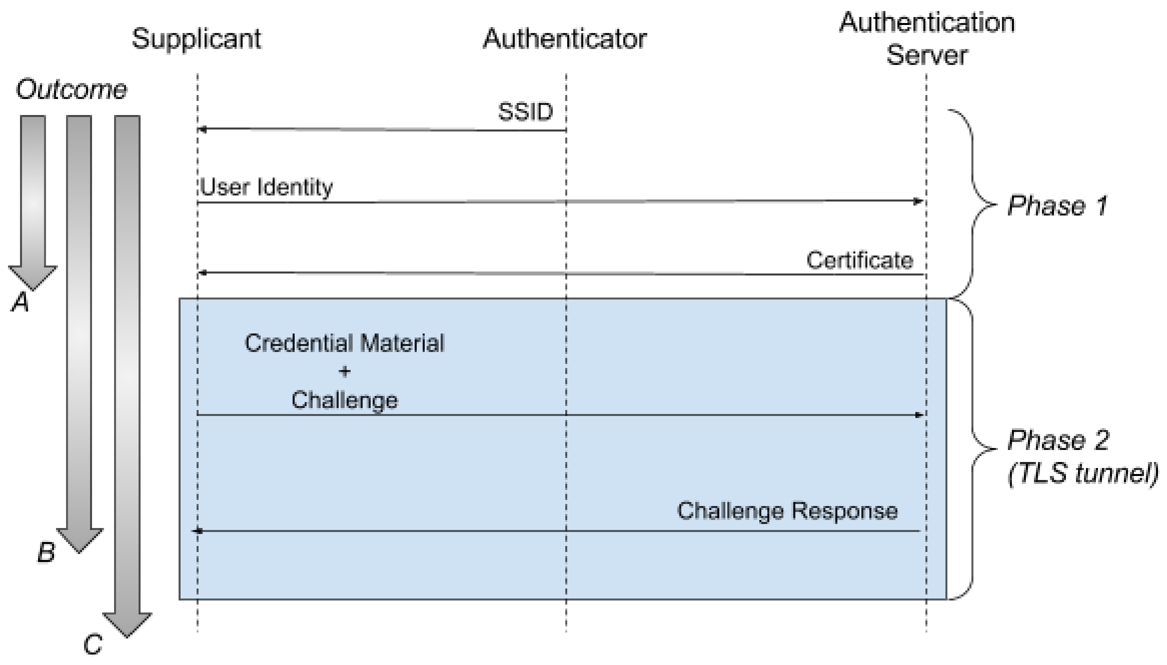

Figure 1: Logical view of EAP-TTLS and PEAP execution, i.e., the drawing outlines the information exchange rather than the actual message exchange. Traffic between Supplicant and authenticator occurs on a wireless link. The "Outcome" arrows indicate the logical points in which the Supplicant may terminate execution: A, when the Supplicant does not validate the certificate; B, when the Supplicant has sent credential material but has not authenticated the Authentication Server; C, when the Supplicant completes the protocol execution successfully.

2.2 Attack objective

We consider attacks aimed at eliciting credential material, i.e, tricking the Supplicant to execute the authentication protocol with an ET. Network credentials are highly valuable because in many organizations the same credentials unlock access to all internal services. For example, at our University, access to the wireless network, to the web application for grading exams, to e-mail, to the web application for management of research projects and budgets and so on, are all unlocked by the same set of credentials.

We are concerned with attacks executed by an ET that remains in close proximity of a Supplicant for a short time, while the Supplicant is not connected to other wireless networks. The typical scenario for these attacks consists of users which carry devices configured for connecting to the wireless enterprise network automatically and move across regions not covered by that network, in particular, outside of the enterprise. Techniques for enabling an ET to connect to Supplicants even in regions in coverage of the enterprise network can be found, e.g., in [ARKN13]. The cited work includes the design and implementation of a jamming device with a cost of around 3500 USD and capable of neutralizing the signal of the legitimate access points for the targeted Supplicant.

We remark that the objective of our attack is *not* enabling the ET to operate as a man-in-the-middle, i.e., inspecting and possibly modifying the traffic to/from the Supplicant. Man-in-the-middle attacks require Internet connectivity and, most importantly, they generally require the ability to engage the user in a working session. Such attacks do not fit the scenario of our concern, in which a device remains in close proximity with an ET for a very short time and potentially anywhere.

To clarify our scope further, we consider the classification of ET scenarios proposed in [LPPE14]: *Replacement*, when the legitimate access point is switched off and replaced by the ET at the same location; *Coexistence*, when the ET coexist with a legitimate access point at the same location; *Remote clone*, when the ET is at a location where there is no legitimate access point; *Ad hoc clone*, like remote clone except that in this case the ET claims to be any SSID at which Supplicants attempt to connect. Although the attacks that we consider could be suitable also for the Replacement and Coexistence scenarios, in this work we focus only on the Remote clone and Ad hoc clone scenarios, with special emphasis on ETs placed at locations that have nothing to do with the enterprise.

Once a Supplicant has initiated an execution of the authentication protocol, actual transmission of credential material to the Authentication Server may or may not occur depending on whether the Supplicant validates the certificate received at the end of Phase 1. In detail, the possible outcomes of an execution of the authentication protocol may be categorized as follows (see the “Outcome” arrows in Figure 1):

A. Supplicant does not validate the certificate, hence it terminates execution without sending any credential information.
B. Supplicant validates the certificate, sends credential information within the TLS tunnel but terminates execution without authenticating the Authentication Server.
C. Supplicant completes both Phase 1 and Phase 2 successfully.

When a Supplicant enters in range of an ET, the Supplicant may or may not initiate an execution of the authentication protocol depending on several factors, including whether the ET is advertising a SSID that triggers automatic connection at the Supplicant, the signal strength of the ET and the presence of other signals for the same SSID from other access points. The scenario of our concern corresponds to an ET placed outside of the enterprise, in which case user devices with the wifi interface active will most likely initiate an execution of the authentication protocol. The outcome of the execution may be either A, or B, or C, as discussed in the next section. We will consider an attack successful when the outcome is either B or C. In these cases the attacker may have obtained either a password or a hash of the password, depending on the protocol configuration and on the Supplicant. Of course, if the attacker has obtained only a hash of the password, the attacker will have to execute a further offline guessing. Whether the attacker will manage to actually obtain the password will depend on the password itself [B12], on the guessing strategy [DZWYH16,JYHHLB17] and on the amount of computing resources devoted to the guessing [ARKN13]. However, we believe that the pessimistic standpoint is more appropriate to our analysis and chose to consider the attack successful even if the attacker has obtained only a hash of the password.

Note that the requirements for an ET able to execute these attacks may be satisfied quite easily: Internet connectivity is not required; battery power may be used; credential material may be stored on a flash card (the ET may disconnect from the Supplicant immediately after completing the authentication protocol).

3 What can be obtained with a simple ET?

In order to provide a concrete evaluation of the risk level associated with widely used Supplicants, we executed a systematic campaign aimed at: (i) assessing the practical feasibility of simple attacks with widely available software and cheap equipment; and, (ii) determining the behavior of different operating systems when exposed to an ET pretending to be the enterprise SSID. We worked in collaboration with the technical people responsible for the networking infrastructure of our University, with the objective of assessing our IT security.

In detail:

- We implemented an ET on a Raspberry PI running Kali Linux (cost approx. 40 euros) and a wireless USB adapter (cost approx. 10 euros) (Figure 2). We used hostapd-wpe and freeRadius-WPE, free packages available in Kali Linux which may be configured to implement the Authenticator and Authentication Server of IEEE 802.1X and log the credentials of any user with which they interact.
- We executed our experiments in a region not covered by the legitimate network, in order to replicate the attack scenario in which the ET is located outside of the enterprise.
- We considered only Supplicants which were exposed to the ET *after* having connected to the legitimate network, in order to make the analysis more realistic. In other words, Supplicants have had the opportunity of receiving the full legitimate certificate chain of the Authentication Server. Unless specified otherwise, all Supplicants were under our full control.
- We examined both Supplicants with and without the correct network profile installed. In the latter case, we attempted to emulate the approach that is simplest to users. To this end, we provided the Supplicant with the minimal information necessary for connecting, i.e., username and password without any further configuration information. With Linux and Android Supplicants the default configuration made connection impossible, thus in these cases we choose the only path available through the user interface that enables connecting, which consists in checking the checkbox “do not validate the certificate”[3].
- We considered Supplicants running the following operating systems: Android 4.4.4, 5.1, 6, 7; Linux Debian 8, Linux Ubuntu 16; Windows 7, Windows 8, and Windows 10; MacOS 11.6; iOS 10.3.3.

[3] In Android 7 there is an alternative and equally simple path which consists in “validate certificate with system certification authorities”. The results obtained in this case are identical to those described in the following.

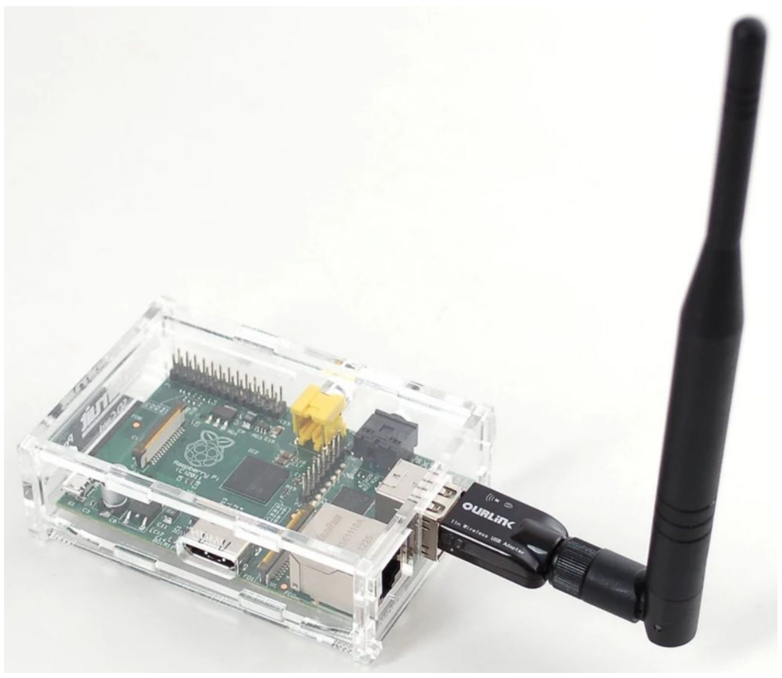

Figure 2: Hardware platform implementing the evil twin used in our experiments (the battery is omitted).

The legitimate network at our University is based on PEAP-MSCHAPv2, is part of eduroam and its SSID is `eduroam`. The certificate chain for the Authentication Server is composed of 3 certificates:

- C1: a self-signed certificate of a certification authority included in the trust store of all major operating systems (Digicert);
- C2: a certificate for an intermediate certification authority (GÉANT's Trusted Certificate Service);
- C3: a certificate for the Authentication Server.

Supplicants are supposed to install the correct network profile for binding this chain to the `eduroam` SSID by executing an application developed by the eduroam consortium and available for each major operating system (https://cat.eduroam.org/). This application parses a configuration file provided by each organization to their users, usually to be downloaded from a web page.

Unless differently stated, we equipped the ET with a certificate chain C1-C2-C3’, that is, a chain differing from the legitimate one only in the last certificate. Obtaining C3’ requires proving possession of a domain name within one of institutions affiliated to eduroam, which is not particularly difficult to achieve. In other organizations the certificate chain is rooted at a self-signed certificate of a certification authority managed by the organization itself, which is also the preferred setting for eduroam. This case will be discussed in the next section.

We summarize our results in the next section. For ease of presentation, we categorize the behavior of Supplicants depending on whether they send credential material to the ET:

- Secure, when the Supplicant does not validate the certificate (the outcome of an authentication protocol execution is A, see Figure 1).
- Insecure, when the Supplicant sends credential information (either B or C in Figure 1).
- UserInsecure, when the user is presented with a binary choice of the form “proceed and connect” vs “abort and disconnect”, the outcome depends on the actual choice and one of the choices leads to sending credential information (either B or C in Figure 1).

Executions that were either Secure or Insecure complete automatically and do not require any user involvement. We will not specify the Android, Linux or Windows version if the corresponding result applies to all the versions that we have tested.

3.1 Supplicant behavior

We consider Supplicants not configured with the correct network profile first. We executed a suite of experiments with the ET configured to use an authentication protocol different from the one of the legitimate network, as follows. As observed above, PEAP is a compound protocol that accommodates several authentication protocols to be executed within the TLS tunnel during Phase 2. A very common option, used in our University as well, is PEAP-MSCHAPv2, in which the MSCHAPv2 protocol is used. This protocol provides mutual authentication and the Supplicant sends a hash of the user password within the tunnel. Another option is PEAP-GTC, in which the GTC (Generic Token Card) protocol is used. This protocol does not provide mutual authentication and the Supplicant sends user credentials in clear text within the tunnel. These credentials are meant to be obtained by the user from a token card or device in response to a challenge sent by the server [ABVCL04]. We configured the ET to use PEAP-GTC. The results, summarized in Table 1 and discussed below, are similar to the finding in [SH13].

Operating System	**PEAP-GTC (Clear Text Password)**	**PEAP-MSCHAPv2 (Hashed Password)**	**Correct Profile PEAP-MSCHAPv2 (Hashed Password)**
Android	Insecure	Insecure	Secure
Linux	Secure	Insecure	Insecure
Windows 7	Secure	Secure	Secure
Windows 8	Secure	UserInsecure	Secure
Windows 10	Secure	UserInsecure	Secure
macOS	UserInsecure	UserInsecure	Secure
iOS	UserInsecure	UserInsecure	Secure

Table 1: Summary of behavior for Supplicants of various operating systems that are exposed to an ET after having been connected with the legitimate network.

Android leads to Insecure executions and sends the password *in clear text* without any user interaction. Both iOS and macOS lead to UserInsecure executions, i.e., the Supplicant sends the password *in clear text* but only if the user actively decides to connect. Furthermore, with these three operating systems, when the Supplicant sends the password it also completes the authentication protocol execution successfully, that is, the Supplicant indeed establishes a network connection to the ET which thus may operate as a man-in-the-middle. We remark that the behavior observed with Android, iOS and macOS is compliant with the GTC protocol and, thus, cannot be attributed to an implementation mistake on the Supplicant.

Linux exhibits only Secure executions because, as it turns out, the Supplicant does not accept using an authentication protocol different from the one used by that SSID earlier (as pointed out earlier, we considered Supplicants that had connected to the legitimate network already). Windows also exhibit only Secure executions because it does not offer native support for PEAP-GTC.

We then executed a suite of experiments in which the ET was configured to use the same authentication protocol as the legitimate network, i.e., PEAP-MSCHAPv2 (Table 1, middle column). The only operating system which always leads to Secure executions is Windows 7. Linux and Android lead to Insecure executions. MacOS, iOS, Windows 8, Windows 10 lead to UserInsecure executions. Furthermore, with MacOS and iOS, the UserInsecure executions would allow MITM attacks, that is, the Supplicant terminates the execution successfully.

Windows 7 implements a form of trust on first use policy in which the Supplicant expects that certain elements of the certificate chain of the first connection be presented again at later connections (the root certificate C1 and the DNS name in C3): since the ET presents a chain without those elements, the Supplicant does not enter phase 2 of the authentication protocol.

MacOS and Windows 10 implement a weaker form of trust on first use: upon receiving a different certificate chain, they ask the user whether to proceed by showing a window with the DNS name of the ET and the SSID of the network. The user may accept or refuse connection with a single click. We observe that, in general, users are aware of the correct SSID but they are hardly aware of the correct name of the authentication server. Windows 8 and iOS also lead to UserInsecure executions but these are less dangerous to users because, in this case, the Supplicant may connect to the ET only if the user *actively* attempts to connect to a wireless network and then selects the SSID of the ET. Then, the Supplicant shows a window asking the user to either accept or refuse to connect, as above.

We now consider Supplicants configured with the correct network profile (Table 1, last column). All operating systems except for Linux lead to Secure executions, while Linux leads to Insecure executions. Furthermore, Linux leads to Insecure executions with *any* certificate issued by the root of the legitimate chain (thus a certificate that can be obtained very easily).

Next, we executed several experiments with the ET equipped with a certificate chain different from the one considered in the previous experiments. We omit the details for brevity and report only the most relevant results. Linux and Android Supplicants that are not configured with the correct network profile, lead to Insecure executions in a variety of cases for the ET certificate:

1. certificate issued by the root of the legitimate chain (thus a certificate that can be obtained very easily);
2. self-signed certificate;
3. certificate issued by a certification authority different from the root of the legitimate chain but included in the default store of the operating system.

We note that, from an attacker point of view, all the three cases are simple to implement. Supplicant behavior with the other operating systems follows the same pattern as in the other experiments. Windows 7 implements a form of trust on first use policy, thus the Supplicant does not enter phase 2 of the authentication protocol whenever the certificate chain presented by the ET does not include the root certificate C1 and the DNS name in C3. The resulting behavior may thus be categorized as Secure. MacOS, iOS, Windows 8 and Windows 10 all implement a weaker form of trust on first use thus, upon receiving a different certificate chain, they ask the user whether to proceed by showing a window with the DNS name of the ET and the SSID of the network. The resulting behavior may thus be categorized as UserInsecure.

Finally, we focussed on the scenario in which the organization runs a certification authority internally and anchors the certificate chain at that authority, which is also the preferred setting for eduroam. In this case, the behavior is as follows.

- Supplicants that are not configured with the correct network profile exhibit the very same results as above: it suffices to configure the ET with a certificate that is either self-signed or issued by a certification authority included in the default store of the operating system.
- Supplicants that are configured with the correct network profile also exhibit the same results as above, i.e., all operating systems lead to Secure executions except for Linux. The requirement for triggering Insecure executions on Linux would be more difficult to satisfy because in this case the attacker should configure the ET with a certificate issued by the

organization itself (the attacker should prove the need of binding a name of the organization to a key and then use the certified name for an ET).

3.2 Targeted attacks

In order to assess the practical feasibility of eliciting credential information from a *specific* target, we executed several experiments outdoor, in a region not covered by the legitimate enterprise network. We considered an idle attacker with an ET in a bag and a walking (voluntary) target with a Supplicant in his pocket. The Supplicant was not configured with the network profile and had connected with the legitimate network earlier, like in the previous section.

When the ET was within 35 meters from the target, the Supplicant always leaked credential material to the ET. Attacks from larger distances did not succeed. We noticed that up to 30 meters every connection attempt resulted in a successful (to the attacker) execution of the authentication protocol, while in the range 30-35 meters some connection attempts failed but the ET managed to trigger a successful attempt in a few seconds nevertheless.

Next, we considered a (voluntary) target driving a car with a Supplicant placed on a seat. The car passed in front of the idle attacker, with an ET in his bag, slowly, at a speed within city limits. The ET never managed to elicit credential information. A short car stop within a range of around 30 meters from the attacker, though, was enough to leak credential information. Whether car windows were open or closed had no effect on the outcome.

3.3 Non-targeted attacks

The previous experiments have been made in a controlled environment with Supplicants under our full control. We executed an experiment in a real environment with a version of the ET carefully modified to not place any user data at risk. Specifically, we modified the sources of the logging module of freeradius-wpe in order to log only the following information.

- For each successful attack: (i) time instant; (ii) hash of the username; (iii) hash of the institution (eduroam user identifiers take the form username@institution); (iv) a flag telling whether a password (GTC) or a password hash (MSCHAPv2) is obtained; (v) only in case the institution is our University, a flag telling whether the credentials correspond to a student or to a staff unit (such information can be derived from the username syntax). No further information was logged, in particular, neither usernames, nor passwords, nor password hashes were logged.
- For each execution of the authentication protocol that terminated after receiving the user identity but before establishing the TLS channel (Figure 1): (i) time instant; (ii) a flag telling whether the Supplicant explicitly rejected the certificate sent by the ET or terminated the execution abruptly. The reason for the latter may include silent rejection of the certificate but also other reasons, including unstable network conditions and inability of the ET to sustain a spike in the number of concurrent executions of the authentication protocol.

We did not attempt to track the real identity of people by other means. The log was stored on a flash memory card on the Raspberry executing the ET and was destroyed after summarizing its content as described below.

We placed the ET within a bag and spent a few hours at a bus stop near our University, which is not covered by the legitimate network, and roaming around our campus. We found 554 different user identities in the log, categorized as follows: 200 with credential material (36.1%); 89 indicating explicit rejection by the Supplicant (16.1%); 265 indicating abrupt termination (47.8%). Credential material is summarized in Table 2.

It can be seen that we gathered 124 credentials in clear text, which would have enabled us to take the identity of the corresponding credentialed users without any effort, wherever username and password suffice to this purpose (as explained above, the ET received username and institution name in the clear, but we stored this information in hashed form for privacy reasons). We also gathered 76 MSCHAP-vs hashes: taking the identity of the corresponding credentialed users would have required the additional effort of guessing the corresponding passwords (Section 2.2).

We emphasize again that the attack did not require any action from the users. Credentials in Table 2 were extracted from users that: (i) happened to be within a few meters from a person with an ET in his bag; and, (ii) were carrying a device which had connected to eduroam in the past and was not configured correctly; and, (iii) the wifi interface of the device was switched on and configured for connecting to eduroam automatically. It is fair to claim that attacks of this kind are indeed a practical means for obtaining enterprise credentials.

	Clear text		**MSCHAP-v2 hash**		**Total**	
University student	101	18.2%	62	11.2%	163	29.4%
University staff	14	2.5%	10	1.8%	24	4.3%
Other institutions	9	1.6%	4	0.7%	13	2.3%
Total	**124**	**22.4%**	**76**	**13.7%**	**200**	**36.1%**

Table 2: Summary of credential material. Percentage values are computed with respect to the total number of authentication protocol executions for which credential material were logged.

To place the figures in Table 2 in perspective, we note that the number of different user identities of our University that connected to the legitimate network in the timeframe of our experiment was around 2700, i.e., we managed to elicit credential material from about 7% of all users by just roaming a few hours around. It is also interesting to note that, according to the content of our log, the number of Supplicants that are configured correctly may be estimated in the range 15%-47%. While we currently have no elements for making this estimate more accurate, we believe the real value should be much closer to 15% than to 47%.

3.4 Discussion

The behaviour observed in our experiments is summarized in Table 1. We focus on Supplicants that are not configured correctly. With Android, an ET may elicit transmission of passwords in clear text without any involvement of the user and in less of a second. Android is used in smartphones, tablets and smartwatches, which are usually always on, with the wifi interface enabled and configured to connect automatically to the SSID of the enterprise network. These devices may thus fall prey of a successful attack virtually anywhere and anytime (at least in regions not covered by the legitimate network). The experiments in the previous section corroborate this claim. This risk level is certainly extremely high and, in our opinion, it is not emphasized adequately in the technical guidelines for users of wireless enterprise networks. As a side note, we note that the market share of Android for mobile operating systems is approximately 86% (Q1-17 https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/).

The behaviour of Linux is very similar to that of Android, but we believe that in this case the risk level is smaller. While Android Supplicants may be tricked into executing the authentication protocol automatically and anywhere, Linux Supplicants are typically notebooks hence they are likely to execute the authentication protocol only when the user is engaged in a working session. An ET should thus be placed where users expect to be able to connect and work. Consequently, attacks executed outside of the enterprise are much less likely to succeed. Furthermore, an ET placed within the enterprise could be detected by the network administrators more easily, at least in principle.

The risk level of Windows, macOS, iOS is, broadly speaking, similar: successful attacks always require a form of user involvement. In this respect, the risk level is relatively small because of reasons similar to those of Linux: successful attacks may occur only when and where the user expects to be able to connect.

4 Countermeasures

A classification of countermeasures to ET-based attacks for the broader category of wireless local networks, i.e., even for those that do not implement WPA2 Enterprise authentication, can be found in [LPPE14]. Typical approaches aim at detecting an ET in operation by uncovering forms of traffic anomalies at various levels of the network stack. With respect to the specific scenario of our interest, though, such countermeasures are not effective as they cannot prevent Supplicants to initiate an authentication protocol execution with an ET. Most importantly, detecting an ET that may be placed virtually anywhere, i.e., even outside of the targeted organization, is extremely hard.

We may identify only two technical countermeasures that can be taken in the short term to mitigate the risks described in the previous sections. First, multi-factor authentication should be deployed for enterprise services. Second, the password policy should allow only passwords that provide a strong resistance to offline guessing attacks. Although these measures are highly useful in general, they are made even more desirable by the fact that, as it turns out, obtaining credential information with an ET is both simple and attractive to attackers. We remark that these countermeasures can only mitigate the effects of a successful attack. There is no way for making sure that Supplicants will not leak credential material to an ET, other than configuring the correct network profile on the Supplicant before connecting.

A further option that, in principle, could be considered is the usage of different passwords for network access and for access to the other enterprise services. This choice would be a step back from SSO and would certainly introduce additional complexity and inconvenience. On the other hand, it would also have significant security advantages. First and foremost, the value of network credentials for attackers would diminish, thereby making the attacks considered in this work less attractive. Second, it would introduce an important compartmentalization in the enterprise architecture. Overall, though, we believe that the resulting hassle to IT staff and users would be too high to make this option practically viable.

More effective countermeasures would require significant changes in protocols, Supplicants, access points and authorization servers. Leaving technical details aside, the number of different actors and interests involved would make such an endeavor extremely difficult. For example, an interesting proposal that did not gain much traction consisted in using certificates for binding the SSID of the network to the public key of the Authorization Server, much like HTTPS certificates bind DNS names to the public key of the corresponding web server [BCT11]. Such an approach would enable an out of the box Supplicant to securely connect to a specified SSID. On the other hand, an actual implementation of the proposal would have required the definition of a global namespace for SSID, the cooperation of certification authorities and significant modifications to networking devices. Another proposal suggested to formalize a form of trust-on-first-use policy: assuming that the first connection with a given SSID is established with a legitimate Authorization Server, subsequent connections with that SSID may only be established with an entity with the same public key [GBLMS10]. Indeed, security policies based on trust-on-first-use have been recently proposed for HTTPS [EPS15]. On the other hand, in addition to the problem of upgrading the networking infrastructure, there is a crucial and difficult to solve problem in this area: how to enable the legitimate server to modify its keys prior to the planned expiration date without incurring the risk of becoming isolated from many of its clients.

The Wi-Fi Alliance has recently proposed a standard for enabling secure, cellular-style roaming for wireless networks that is built upon WPA2 Enterprise [WF16]. Public hotspots belong to a global infrastructure potentially distributed around the world. Supplicants may authenticate to hotspots either with a SIM, or with a certificate, or with a username/password pair. The configuration problem for out of the box Supplicants that authenticate with username and password has been solved by assuming that all “Passpoint certified” Supplicants: (i) will be equipped out of the box with the certificate of a certain certification authority specified in the standard; and, (ii) will connect to Passpoint-enabled networks only when receiving a certificate chain whose root is that authority. It remains to be seen whether this approach will be really effective and how Supplicants that are not Passpoint certified will be handled. Indeed, commercial implementations of such cellular-style wifi roaming are already available and in those cases Supplicants may connect only after installing a dedicated *application* (see, e.g., http://www.ipass.com).

4.1 A more radical proposal

A direction that, to the best of our knowledge, has not yet been explored consists in modifying the model of enterprise wireless networks presented to users. Today the Supplicant presents a list of in range SSIDs; the user selects the SSID of interest and the Supplicant connects to that SSID; the user may instruct the Supplicant to connect again to a specified SSID automatically. This model applies to all kinds of wireless networks: open networks, networks protected with WPA Personal (i.e., the same password is shared by all devices) and networks protected with WPA2 Enterprise). We propose to investigate a different framework, as follows.

The Supplicant maintains *two* separate lists of networks with which automatic connection is possible: one for open networks and for WPA Personal networks; another for WPA2 Enterprise networks. The former is managed as it is today, while the latter is managed with the following rules:

1. Networks in the list are described by a pair <SSID, AS name>. The list cannot contain multiple pairs with the same SSID.
2. The Supplicant may connect automatically with a WPA2 Enterprise network only if the network is in the list.
3. The list is initially empty. Insertion and removal may be done only by the user. Insertion may occur in only two ways:
    a. the user specifies SSID and AS name (for example, by typing in a form in a GUI of the Supplicant); or,
    b. the user asks the Supplicant to explore the networks in range and selects one of the SSIDs; the Supplicant initiates the authentication protocol, obtains the AS name and asks the user whether the corresponding <SSID, AS name> pair should be inserted in the list.
4. Certificate validation can never be skipped.
5. Certificate chains rooted at a self-signed certificate that is not in the store of the Supplicant are treated as follows: the root certificate must be accepted by the user upon inserting the SSID in the list; later connections with the same <SSID, AS name> pair must occur with the same chain; and, the root certificate can be used only for that pair, i.e., it cannot be used for certifying the chain of other enterprise networks or of other entities.

As an example, consider an out of the box Supplicant. The Supplicant could not connect automatically with any enterprise wireless network. Configuration for the enterprise network at our University, with SSID `eduroam` and Authorization Server named `raggio.units.it`, would occur with procedure 3-a (procedure 3-b will be discussed later). When in range of the SSID `eduroam`, the Supplicant would then start the WPA2 Enterprise authentication protocol automatically but would enter Phase 2 of the protocol only upon receipt of a valid certificate for the DNS name `raggio.units.it`.

In this new scenario, an ET could receive credential material only in these cases: (i) the Supplicant enters in range of the ET *before* connecting with the legitimate network; and (ii-a) the ET has a self-signed certificate accepted by the user; or, (ii-b) the ET has a chain rooted at a certification authority issuing fraudulent certificates. The resulting scenario would not be foolproof, but the bar would be much higher than it is today.

Indeed, the resulting scenario would be aligned to the higher security level currently available for HTTPS. Consider an out of the box device. A user that knows the name of an HTTPS web server may connect the device to that server securely: the only risks would be (i) and (ii-a)/(ii-b) above. Out of the box devices can thus be considered as being *secure by default* with respect to HTTPS usage, in the sense that they do not need any specific configuration for connecting to a named server securely. In contrast, with current technology, a user that knows the name of a WPA2 Enterprise network (i.e., its SSID) *cannot* connect the device to that network securely. The proposed approach, thus, would provide an uniform and more secure framework for out of the box Supplicants. Figure 3 shows a possible graphical interface for the proposed model.

Key advantages of the proposed approach are:

- Out of the box Supplicants would be secure by default, in the sense just described, with respect to WPA2 Enterprise wireless network connection.
- Modifications would be needed *only* on the Supplicants, at least in principle. No changes on networking devices or protocols would be required.
- No changes on certificate structure or management would be required either: Supplicants would use the very same rules and certificate types that are used today for binding a public key to a server name.
- Deployment and upgrade could be incremental: Supplicants offering the new user interface (and stronger security) could co-exist with traditional Supplicants.
- It would accommodate existing large-scale roaming services such as eduroam and Passpoint without any change, at least in principle. The reason is because Supplicants establish a TLS tunnel with the Authorization Server of their home institution, as identified in the early stage of the authentication protocol, irrespective of their actual location.

The description so far assumes that users connect only through procedure 3-a, which requires users to know the name of the Authorization Server in advance. Satisfying this requirement could be exceedingly difficult for some organizations or for some users. For this reason, procedure 3-b makes it possible to connect even in the way familiar to the user, by exploring the networks in range and selecting the desired SSID. This option clearly introduces a further risk, i.e., an ET impersonating the desired SSID could receive credential material also in case: (i) the Supplicant enters in range of the ET *before* connecting with the legitimate network; and (ii-c) the ET has a valid certificate for a name different from the name of the legitimate AS.

We emphasize that even with procedure 3-b the resulting scenario would be arguably more secure than it typically is today because the Supplicant would have to enter in range of the ET and to actively decide to connect *before* having connected to the legitimate network. Furthermore, in that case, the Supplicant would not be able to connect to the legitimate network and thus would have no wireless connectivity within the enterprise (unless the Supplicant remained permanently in coverage of the ET). This fact would make it possible, at least potentially, to realize that something wrong has occurred. Note also that the network list would contain the name of the AS of the ET, which would certainly be different from the name of the legitimate AS because the ET exhibited a valid certificate for its (fraudulent) AS.

Finally, we observe that Procedure 3-b is not an essential component of the proposed approach. Procedures 3-a and 3-b support two different security/usability trade-offs, in which the latter is simpler to use and more similar to the interaction model currently used. While the proposed approach would constitute a significant departure from a simple and established model presented to users, we believe the resulting security improvement make the proposal worth investigating.

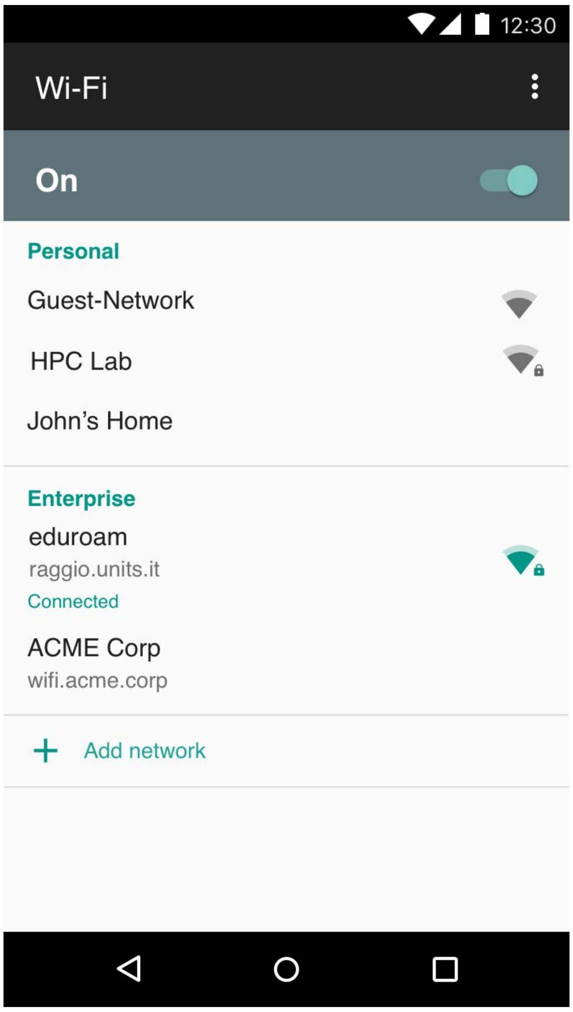

Figure 3 A possible interface for the proposed model. This screen displays the wi-fi networks with which automatic connection is possible. WPA2 Enterprise networks are clearly separated from the other networks and identified with explicit visibility for the AS name. The “add network” functionality, not shown, shall also handle WPA2 Enterprise networks differently from the other networks, as described in the main text.

5 Conclusions

WPA2 Enterprise Supplicants that authenticate with username and password and that are configured with the minimal options necessary for connecting to the wireless enterprise network, are at risk of leaking credentials virtually anywhere and anytime. We have shown that this risk is not a mere theoretical possibility. On the contrary, it can be exploited easily, cheaply, quickly and with high probability of success. Convergence of organizations toward SSO architectures in which a single set of credentials unlock access to all services of the organizations, coupled with the ubiquitous availability of wifi-enabled personal devices which often contain enterprise credentials, have made attacks of this sort particularly attractive to attackers and hard to detect.

It would be easy to dismiss our concerns as well known technical issues, or as an obvious consequence of wrong user behavior, or as an unsurprising side effect of ineffective IT management policy, or as yet another of the many security vulnerabilities of today's technology. We believe such an attitude would be wrong: people are not supposed to be aware of technical issues of this sort and we, as a technical community, should definitely provide them with devices that are secure even out of the box. In our opinion, a technology that in many, if not most, of its practical deployments puts enterprise credentials at risk is no longer acceptable.

Solving the corresponding problems in order to provide solutions that are both practical and more secure is much easier said than done, because such an effort would require commitment and involvement of a number of different actors, usually with competing interests. However, the community should at the minimum acknowledge that there is a security problem in a fundamental technology and express loudly that magnitude and extension of this problem are underestimated.

Acknowledgments

The authors are very grateful to the networking staff of the University of Trieste for their invaluable help and technical advice.

References

[ABVCL04] “Extensible Authentication Protocol (EAP)”, Aboba B., Blunk L., Vollbrecht J., Carlson J., Levkowetz H. RFC 3748, June 2004.

[ARKN13] "A Practical, Targeted, and Stealthy Attack Against WPA Enterprise Authentication.", Cassola A., Robertson W., Kirda E., Noubir G., NDSS. 2013.

[B12] "The science of guessing: analyzing an anonymized corpus of 70 million passwords.", Bonneau J., IEEE Symposium on Security and Privacy 2012, pp. 538-552.

[BCT11] “Secure Open Wireless Networking”, Byrd C., Cross T., Takahashi T., Black Hat 2011.

[BL16] “2016 Enterprise Mobility Survey Results: Strategic Imperatives”, Bassett B., Lund D., IDC Web Conference 2016.

[C16] “BYOD and Mobile Security Report”, Crowd Research Partners, 2016.

[EPS15] “Public Key Pinning Extension for HTTP”, Evans C., Palmer C. Sleevi R., RFC 7469, April 2015.

[DZWYH16] "Targeted online password guessing: An underestimated threat.", Ding W., Zhang Z., Wang P., Yan J., and Huang S., Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp. 1242-1254. ACM, 2016.

[FEOS07] “Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i”, Frankel S., Eydt B., Owens L., Scarfone K., NIST Special Publication 800-97, February 2007.

[GBLMS10] “Practical Defenses for Evil Twin Attacks in 802.11”, Gonzales H., Bauer K., Lindqvist J., McCoy D., Sicker D., IEEE Globecom Communications and Information Security Symposium, December 2010.

[J12] “BYOD–is it Good, Bad or Ugly from the User Viewpoint?”, Jones J, Microsoft Secure Blog, July 2012.

[JYHHLB17] “Zero-Sum Password Cracking Game: A Large-Scale Empirical Study on the Crackability, Correlation, and Security of Passwords”, Ji S., Yang S., Hu X., Han W., Li Z., Beyah R. IEEE Transactions on Dependable and Secure Computing, vol. 14, n. 5, September/October 2017, pp. 550-564.

[LPPE14] “Undesired relatives: protection mechanisms against the evil twin attack in IEEE 802.11”, Lanze F., Panchenko A., Ponce-Alcaide I., and Engel T., Proceedings of the 10th ACM symposium on QoS and security for wireless and mobile networks (Q2SWinet '14).

[M13] “Wireless PEAP-MS-CHAPv2 Authentication Could Allow Information Disclosure”, Microsoft Security Advisory 2876146, August 2013.

[MW12] “eduroam Policy Service Definition (version 2.8)”, Milinovic M., Winter S., GN3-12-192, February 2012.

[S16] “Syntonic 2016 Employee: BYOD Habits and Attitudes”, Syntonic, Fall 2016.

[SH13] “BYO-Disaster and why corporate security still sucks”, J. Snoodgrass, J. Hoover, DEFCON 21, August 2013.

[SS12] “Guidelines for securing wireless local area networks (WLANs)”, Souppaya M., Scarfone K., NIST Special Publication 800-153, February 2012.

[VP17] “Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2”, M. Vanhoef, F. Piessens, Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17).1313-1328

[Y13] “BYOD PEAP show”, Josh Yavor, DEFCON 21, August 2013.

[Y16] “Results of implementing WPA2-enterprise in educational institution”, Yanson K., IEEE 10th International Conference on Application of Information and Communication Technologies, 2016.

[WF16] “Wi-Fi CERTIFIED Passpoint™ (Release 2) Deployment Guidelines Rev 1.1”, Wi-Fi Alliance, December 7, 2016.

[WWW15] “The eduroam architecture for network roaming”, K. Wierenga, S. Winter, T. Wolniewicz, RFC 7593, September 2015.