Attacks and vulnerabilities of Wi-Fi Enterprise networks: User security awareness assessment through credential stealing attack experiments

Article *in* Computer Communications · September 2023

DOI: 10.1016/j.comcom.2023.09.031

CITATIONS
14

READS
296

6 authors, including:

Ivan Palamà
Consorzio Nazionale Interuniversitario per le Telecomunicazioni
23 PUBLICATIONS **140** CITATIONS
SEE PROFILE

Francesco Gringoli
University of Brescia
154 PUBLICATIONS **3,103** CITATIONS
SEE PROFILE

Giuseppe Bianchi
University of Rome Tor Vergata
393 PUBLICATIONS **19,616** CITATIONS
SEE PROFILE

Attacks and vulnerabilities of Wi-Fi Enterprise networks: user security awareness assessment through credential stealing attack experiments

Ivan Palamà[a], Alessandro Amici[a], Gabriele Bellicini[b,c], Francesco Gringoli[b], Fabio Pedretti[b], Giuseppe Bianchi[a]

[a] *CNIT/University of Rome "Tor Vergata"*
[b] *CNIT/University of Brescia*
[c] *VerXo srl, Italy*

Abstract

Enterprise Wi-Fi networks are essential for businesses and public administrations as they provide a perfectly scalable and secure system. In the university environment, they are often deployed to offer services to students. One of the most famous university Wi-Fi Enterprise networks is Eduroam, which stands for education roaming; it is a worldwide Wi-Fi access and roaming service widely adopted by the international research and education community. It is based on 802.1x mechanisms that use TLS tunnels for achieving mutual authentication goals, and, as such, it requires careful configuration of mobile devices and responsible users' behaviors to avoid trivial attacks carried out with rogue Access Points (APs). Differently than employees in a corporate network whose devices are properly configured by ICT teams, the user base of Eduroam consists of (likely) millions of students and professors around the world, with a myriad of different and uncontrolled devices. To assess the security of 802.1x in general, and more specifically that of Eduroam, we ran attacks against two communities of students of increasing size in order to test how users (and their devices) react when rogue 802.1x APs appear in the list of available networks. We then focused our attention on devices, and investigated their detailed dependence on different WPA-Enterprise configurations and certificate settings. The aftermath is that, even with a completely passive attack (users are keeping devices in their pockets), it is possible to steal credentials from more than one-third of the students. While most of the 802.1x vulnerabilities employed in this work should be considered somewhat known (being disclosed in former technical papers), our work appears to raise a threefold concern: i) most pragmatic 802.1x configurations appear to be grossly insecure; ii) no Apple's iPhone felt in our attack unless explicitly forced by the user, owing to its reduced possibility for a user to misconfigure the terminal; and iii) the awareness of Wi-Fi authentication threats even in relatively skilled end users is close to zero.

Keywords: privacy, credential stealing, attack, network, security

1. Introduction

Wi-Fi is one of the most disruptive and catalytic digital communication technologies, enabling the connection of many devices in a completely new way, providing an exceptional user mobility service. This new wireless communication paradigm immediately raised security concerns. Indeed, since all it takes is a wireless network card to eavesdrop on data sent wirelessly, it is clear that data security and user privacy are crucial issues. These security requirements are even more evident in enterprise and public administration contexts where confidential company data and personal user data (e.g., clinical data of hospital patients) may be transmitted over the air. One of the most technically fascinating and challenging projects is undoubtedly the Wi-Fi Enterprise Eduroam network, which since 2002 has aimed to offer a worldwide connection service dedicated to all education and scientific research users. Eduroam allows organizations to easily provide Internet access to their users in mobile conditions thanks to the flexibility of the IEEE 802.1x authentication protocol family. Students, researchers and professors belonging to organizations that are part of Eduroam can use their own credentials when connecting to the network of a visited institution: the visited authentication server is configured, in fact, to securely forward the provided username and password to the home authentication server for local verification. By analyzing the security aspects of such a scenario, it is possible to identify three crucial entities: the Eduroam network, the institutions, and the user's terminal. The security infrastructure of the Eduroam network relies on the use, verification, and proper configuration of the Eduroam root CA certificate within each of the user's devices. Eduroam institutions provide login credentials and should instruct users to configure the network on their devices properly. A Configuration Assistant Tool (CAT) was developed to help organizations offer users access to Eduroam. This tool, available for the most common platforms, allows users to configure their devices quickly and easily using Eduroam profiles configured by organizations. Unfortunately, configuration guides and pre-configured profiles are often outdated or incorrect; e.g., instructions of University of Rome "Tor Vergata" suggest leaving the certificate field at its default value, thus introducing serious vulnerabilities in the network access configuration. In addition, the majority of the tested devices not only do not advertise the users about the risks, but they are also pre-configured with vulnerable profiles. For these rea-

sons, users can unconsciously expose their credentials to attackers that set up rogue access-points, also called Evil Twins, or use some man-in-the-middle mechanism. Such vulnerabilities, however, are not specific to Eduroam: they are rooted in the adopted 802.1x mechanisms, more specifically in the WPA-Enterprise (Wi-Fi Protected Access) authentication methods that, although secure "per-se", can be used in the wrong way. Users can be reckless when prompted with security alerts that are not always easy to understand about some certificates not being trusted by the system, or the device itself may not prompt at all the user. By recreating a malicious 802.1x network that advertises the name of a legitimate one, an attacker can hence aim at stealing users' access credentials. While the experimental analysis of the problem does not necessarily need to be restricted to the Eduroam case, any interesting issue discovered while experimenting with a generic enterprise network and its users would have a much larger impact and resonance if considered on the Eduroam network because of its size. In addition, some Universities configure the wireless access credentials of their employees – professors – by simply mirroring the corporate usernames and passwords used for accessing services like electronic mark sheets: attacking students might, in principle, modify the marks in the system before they are digitally signed. The reported vulnerability becomes a serious problem of identity privacy worldwide since the attacker can identify himself as a university professor and use all the services related to him (e.g., GèANT eduGAIN). Moreover, many universities are now offloading services they were once managing internally, like email and web publishing: having shared users' credentials with large productivity systems like Google or Microsoft allows a malicious attacker to access all services connected to them, thus increasing the severity of the vulnerability.

In this paper we move from such observations and we first set up the attack aimed at capturing the user's credentials in a WPA-Enterprise scenario by describing the Eduroam authentication service, which is perfectly 802.1x compliant. We then run two distinct sets of experiments characterized by decreasing level of control and selection of the groups of attacked users, their skills and their size. Interestingly, the results are quite similar. In general, our results show a remarkable lack of awareness about the risks and the vulnerabilities that affect the WPA-Enterprise systems, including Eduroam. We find perhaps surprising that in front of a non-marginal literature that has documented such vulnerabilities in the course of many past years [1, 2, 3, 4, 5, 6, 7, 8] which can lead to the implementation of many attacks such as Man-in-the-Middle (MitM) [9, 10, 11, 12, 13, 14, 15], Denial of Service (DoS) [16, 17, 18], Key-recovery [19, 20] and Traffic decryption attacks [21, 22], our attack was still successful, pointing out the substantial difference between literary theory awareness and practical awareness in the real world. Mainly driven by the curiosity to understand whether certain brands of mobile phones had a certain "resilience" to such attacks, we ran an experimental analysis designed to understand how different devices react to different forms of rogue AP attacks. We ran our test using three different network certificates (self-signed, expired and valid), four network authentication protocols, and four different certificate control strategies.

The rest of the article is organized as follows: after some necessary background in Section 2, we describe the experimental attacks and results in Sections 3 and 4, where we consider, respectively, a controlled group of users participating in the experiment and a large set of unaware students during their day-to-day activities. We investigate in Section 5 the impact of the device's brand and its configuration on the feasibility of the attack, pointing out how different combinations of these factors have distinct effects also on the user awareness. After reporting the most significant literature close to our work in Section 6, we conclude the paper in Section 7 where we also outline further directions for research.

2. Background

We provide in this Section a quick overview of the Evil Twin attack and 802.1x WPA-Enterprise authentication mechanism adopted inside Eduroam: we focus on the necessary aspects for understanding the vulnerabilities that make the attack possible. We refer interested readers to the standards [23, 24, 25, 26, 27] for further details.

2.1. Evil Twin attack

Evil Twin attack is a well-known [28, 29, 30] wireless security threat that leverages the trust users have in Wi-Fi networks to compromise their data. In this attack, an adversary creates a fake access point (AP) that closely resembles a legitimate one using the same SSID and transmits with a stronger signal. Users cannot differentiate between the legitimate AP and the evil twin AP, assuming both are legitimate APs, in such situations the client will prefer to connect to the one with the strongest signal. As a result, users inadvertently connect to the malicious AP, believing it to be the genuine network. Once connected, the attacker can perform various attacks, such as intercepting and manipulating the users' data, launching man-in-the-middle attacks, or gaining unauthorized access to their devices.

One of the key factors that make Evil Twin attacks successful is the lack of user awareness of the risks associated with connecting to unsecured or unverified Wi-Fi networks. Attackers can exploit this lack of knowledge by making their rogue APs more attractive to users, for example, by offering a stronger signal or disabling encryption mechanisms.

2.2. 802.1x WPA-Enterprise Eduroam authentication

In Eduroam, the authentication is based on the IEEE 802.1x standard for port-based network authentication, which ensures that only authorized users get access. 802.1x includes the usage of EAP (Extensible Authentication Protocol), which allows different authentication methods. Depending on the configured EAP method, i.e., EAP-TTLS, PEAP or EAP-TLS, a secure tunnel from the user's terminal to her/his own institution authentication server is established: this tunnel is used for mutually authenticating users to their own networks, by exchanging public X.509 certificates and users' credentials. IEEE 802.1x

authentication involves three main actors: a supplicant (the mobile terminal), an authenticator (the Access Point), and an Authentication Server. Even though the last two actors usually belong to the same entity that manages the enterprise network, in the Eduroam case the AP belongs to the Service Provider (SP), i.e., the visited institution, and the Authentication Server (AS) belongs to the Identity Provider (IdP) of the user's home institution. While this detail does not actually modify the way the attack is mounted, in the following we focus more specifically on a typical Eduroam scenario. To ensure that users can connect internationally using the credentials provided by their home institutions, Eduroam has a hierarchical infrastructure of linked Remote Authentication Dial-In User Service (RADIUS) servers containing users' data (usernames and passwords) that securely forwards user credentials to the users' home institutions, where they are verified and validated. Eduroam security is based on three trust relationships:

1. The direct trust relationship between the end user and IdP, managed by the user's home organization, established through mutual authentication;
2. The direct trust relationship between IdP and SP, namely the network operator at the visited location, established through the use of the proxy hierarchy of RADIUS servers (organizational, national, global);
3. The transitive trust relation that makes the SP trust the user in order to use its network resources.

The authentication procedure involves two authentication steps:

1. An external layer authentication (EAP-TTLS or PEAP) is performed to establish a secure communication tunnel between the user client device and the home AS;
2. Inside the established tunnel, the supplicant runs an internal authentication algorithm (PAP or MS-CHAPv2) using the credentials provided by his home institution to verify his identity.

In the following of this section, we will refer to Figure 1 which illustrates the authentication procedure executed when a user connects under roaming conditions. In case the attack is run against a corporate network that owns both the AP and the AS, the only difference is that there is no actual AS selection or proxying, and the communication flows directly between the authenticator and the corporate AS.

External authentication: i) To access the network, the supplicant provides to the SP AP the user's identity "username@institution.tld" where institution.tld is the user realm, and the username is optional. ii) The SP AP forwards the identity to the local (visited) AS, which checks if it is responsible for that realm. Since it is not, it transfers the identity to the next RADIUS server at the national or international level according to the realm part of the user's identity, until the user's home IdP AS is found. iii) After verifying the received identity, the user's home IdP AS sends back to the supplicant its certificate. iv) If the supplicant validates the certificate, the secure tunnel is established. The certificate of the IdP AS plays a crucial role in the authentication phase: failure or false verification of the certificate would cause the user serious security problems.

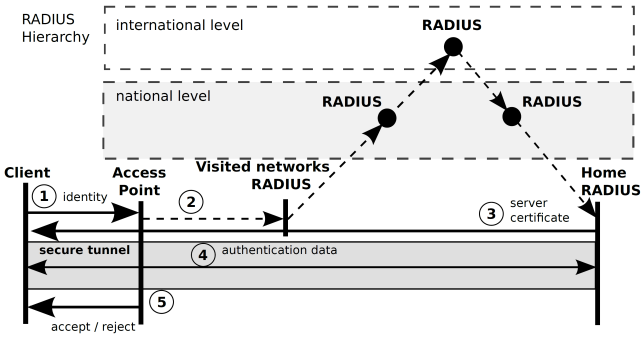

Figure 1: Eduroam Authentication Process, figure taken from [31].

Internal authentication: iv) The supplicant can now use the secure tunnel to authenticate the user to the user's AS: if this phase succeeds, then v) the user's AS can transmit the result to the AP, which will then authorize or deny network access when credentials are wrong.

Authentication vulnerabilities: As mentioned above, the purpose of the external authentication phase is to set up a secure and authenticated tunnel between the supplicant and the remote AS. By reviewing the authentication phase, it can be pointed out that: i) if the user's anonymous identity is not used in the external authentication phase (i.e., username@realm is sent), the user's anonymity is not maintained, and the attacker can link the user's identity with the hardware identity (such as MAC address), thus he can violate the user's privacy (identity and location); ii) Even if the protocols used in the internal authentication phase are secure, they vulnerably rely on the fact that a secure tunnel for the exchange of authentication data has been established in advance, then if the user's client does not validate the certificate provided by the authentication server, the attacker can learn the user's credentials and compromise the user's privacy.

In the next two sections, we describe the two experiments that we run to measure the attack's successfulness in capturing users' credentials. In Section 3 we consider a small group of collaborating users walking through a well-defined path where we are running the rogue AP: their participation allows us to use a questionnaire-based approach for validating our findings. In Section 4, instead, we use three rogue APs located in crowded hallways or classrooms: we use this "in the wild" approach to validate the findings of the previous Section on a larger group of users. Section 5 presents an in-depth experimental analysis of the behavior of some devices with different WPA-Enterprise configurations. Finally, Section 7 draws conclusions and outlines further directions for research.

3. Experiments in the controlled environment

We describe in this Section the experimental attack we carried out on a controlled group of users at the University of Rome "Tor Vergata". Before reporting the collected results, we first describe the hardware and software solutions we used for setting up the rogue AP and the questionnaire we prepared to analyze the users' impressions.

3.1. Experimental setup

The experiment was executed by implementing an Evil Twin attack based on the creation of a malicious AP reproducing a rogue Eduroam network. The setup included the following components:

Hardware: to maximize the flexibility of the attack, we deployed the rogue AP with all the necessary software on an **ASUS ROG STRIX GL703GE** laptop, powered by an Intel i7-8750H 6-Core CPU clocked at 2.2 GHz and embedding an Intel 9260 802.11ac wireless network card. To provide Internet connectivity to victims, we added an external Wi-Fi card, a **TP-Link Archer T3U** USB adapter.

Software: on the laptop we installed an Ubuntu 18.04 Linux distribution, which we configured to provide connectivity to victims by masquerading and routing their traffic over the external USB Wi-Fi stick. To this end, we also set up a DHCP server for providing IP addresses on the same range of the legitimate Eduroam network once victims successfully authenticated to the internal malicious AP in order to recreate a similar environment. To set up the fake Eduroam network, we attached to the internal Wi-Fi card an instance of **hostapd** v2.9, an open-source software that can transform all supported wireless network cards into APs. We configured it to broadcast the Eduroam SSID and to provide 802.1x authentication through a RADIUS server also running on the laptop. For this software, we chose **FreeRADIUS** v3.0, beyond being open-source, it is also adopted by the Eduroam federation. The server was terminating the TLS tunnels established by the victims' mobile devices during the attack and had the ability to collect their Eduroam credentials. Although all participants authorized the experiment, their credentials were hashed with SHA256 to avoid privacy concerns and were promptly discarded. To better manage the attack, we implemented a Python-controlling application that coordinated all the software pieces.

3.2. Phases of the attack

In order to assess the current level of security awareness in a medium-sized university, we engaged in the experiment 35 students in computer and electronic engineering and 2 ICT professors; all participants were from the same institution and used the Eduroam configuration provided by the local institution. The experiment involved the submission of two anonymous pre- and post-attack questionnaires with multiple-choice and open-ended questions to the group of 37 users. The goal of the questionnaires was to determine the degree of awareness of a relatively skilled set of end users before and after falling victim to the attack. The percentages reported in the questionnaires refer to the number of users in relation to the total, so it is possible that the sum of the percentages associated with the answers exceeds 100% in the case of questions with multiple answers. More into detail, each user underwent these three steps:

Fill the Pre-attack questionnaire: The anonymous pre-attack questionnaire consisted of 16 multiple-choice and open-ended questions aimed at analyzing the network-related security skills and threat awareness of a relatively significant sample before they have been under attack.

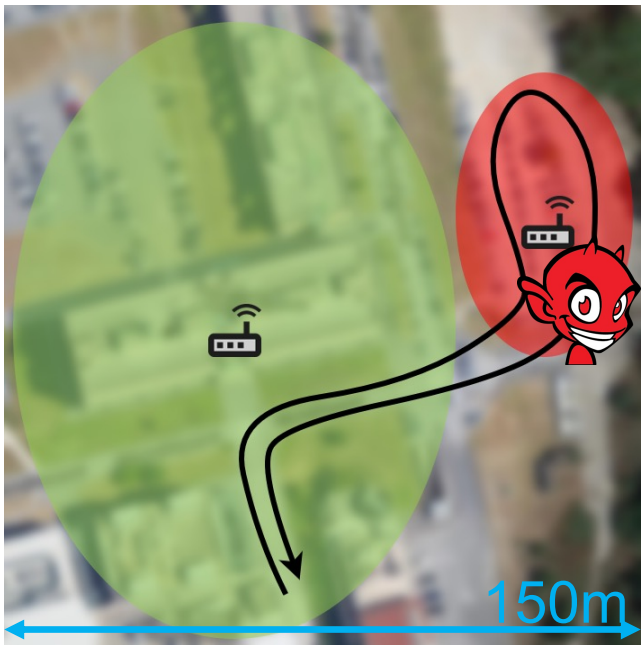

Figure 2: Experiment area. Green: the area of legitimate Eduroam network. Red: the area of the rogue AP. Black arrow: the walking path.

Go through the attack: The attack assumed that users have previously successfully connected to the legitimate Eduroam network. We did not ask them to make any changes to their configurations, reckoning that they have configured the network automatically through the official Eduroam CAT or manually by configuring their credentials (identity and password), authentication protocol, and certificate verification fields. After completing the first questionnaire, users were simply asked to participate in a security test and follow the path shown in Figure 2: this happened without telling them that they would have been attacked. Along the path, the malicious AP was trying to deceive users' mobile devices into establishing a connection. Attack had the chance to be successful thanks to (i) the stronger signal of the AP compared to that of the legitimate network and to (ii) the relatively long time of exposure (the entire path was approximately 300m). Since the user has previously connected to the legitimate Eduroam network, we expected that when her/his device was within the range of the malicious AP, it would automatically try to connect without any user intervention in a passive manner.

Fill the Post-attack questionnaire: After being attacked, samples of students and professors were asked to answer a second anonymous questionnaire with 14 multiple-choice and open-ended questions. The goal here was to understand whether they understood that they were under attack, how they realized that, and how much the awareness of wireless network security threats has increased after realizing they have been attacked. An anonymous id was used to link responses from the two questionnaires.

We analyze next in separate subsections the behavior of the tested devices and try to understand the operating system's dependency and the user's unawareness.

3.3. Results: impact of devices

We report in table 1 the 33 unique mobile devices belonging to the 37 users that received the attack: some participants had the same phone. After analyzing the behavior of smartphones during the experiment, we concluded the following.

Dependence on the OS: Of the 37 participants, 33 used a different smartphone brand/Operating System configuration, with 9 Apple iPhones running iOS, and the remaining running Android. It appears that all Apple iPhones are not vulnerable: they do not connect to the rogue Eduroam AP and do not provide user credentials. It is worth noting that during the attack, users were not interacting with their phones, this might have changed the outcome, as we will see in Section 4 and shortly explain hereafter. Unlike Android smartphones, in fact, Apple smartphones do not allow the user to configure the certificate control policy. When iOS devices connect to the Wi-Fi network, they show the received certificate and ask the user to accept it or not. This behavior leads to the fact that during the connection with the malicious Eduroam AP, the device shows the users the forged Eduroam certificate. This behavior should have allowed users, in principle, to understand what was going on, thereby making them aware of the attack and, in our case, impeding its completion as the user intervention was necessary - indeed, we did not ask participants to interact with their smartphone so as to perform a purely passive attack. By analyzing the experimental results, it can be found that Android smartphones are vulnerable instead (even with the latest Android OS, version 11 during the test). This is because, unlike iOS, Android OS allows users to configure the network in a very detailed way, specifying authentication protocols and certificate control rules. When investigating the certificate control policy configuration in Android smartphones, we found that in several smartphones, users may not choose any policy, and the default policy is to accept unverified network certificates, which makes smartphones really vulnerable to the attacks even passively, without users' intervention as it happened during the experiment. As we will see later, the behavior of Android was modified – better to say, corrected – only recently with the rollout of Android 13 as we describe in Section 5.

Dependence on the Android version: Since all vulnerable smartphones come with Android OS, we ran some additional experiments in a controlled lab in order to identify the reasons behind their vulnerable behavior. We analyzed the different Wi-Fi configuration dialogs exhibited by Android OS from version 5.1 to 13. We found two substantial changes in Android Nougat 7.0 [32, 33] that respectively introduce: i) The display of a warning of potential risks if users do not provide a CA certificate in the EAP configuration. ii) The addition in the EAP configuration menu of the option to not specify an EAP CA certificate and a user certificate; and one important addition in Android 13, the TOFU (Trust on First Use) authentication approach [34] which allows users to trust an enterprise network (EAP) by installing the server's root CA and setting its domain name. TOFU allows the device to obtain a non-authenticated public key when a user connects to a corporate network for the first time and retains the key for subsequent connections. Therefore, it is necessary to emphasize that Android 13 introduced a server certificate validation requirement for corporate networks. This additional requirement created a hurdle for users, who had to install a CA certificate for such networks. TOFU allows users to connect to a PKI-based corporate network simply by accepting its root CA. Their main effect is to require users to make conscious choices about certificate control policy. This leads us to conclude that the vulnerability only exists in devices with an Android version below 7.0. The tested devices were all using newer Android versions, and indeed the majority of them were not vulnerable; however, some others, which even come with more recent Android OS versions, do not display any warning and apply the default vulnerable policy, i.e., they do not check the validity of the received certificates. An example of such behavior is the Xiaomi Mi 10 with Android 10/11. This suggests that the customization of the Android OS performed by the manufacturer may also play a key role in the vulnerable behavior of devices. Possible countermeasures to Android vulnerabilities could include a new default behavior that requests and checks the certificate provided by the network and shows the user an alert if abnormal network behavior is detected (e.g., an AP without a certificate or with a different certificate). It would also be useful to add a link to a network configuration guide to educate the user on possible misconfigurations and their impacts.

Brand	Model	OS
Apple	iPhone 11 Pro	iOS 14.5
	iPhone 11 Pro, 11, XS, 8, 7	iOS 14.4
	iPhone 8	iOS 13.3
	iPhone 8	iOS 13
Samsung	S20, S10, S10e, S9, Note 9	Android 10
	S8, A70	Android 9
	S7	Android 8
Xiaomi	Mi 10, Mi 9T Pro	Android 11
	Mi 10, Mi10T, Mi10 Lite, Mi 9, Mi 9T Pro, Poco X3, Redmi Note 8, Redmi Note 7	Android 10
OnePlus	8	Android 11
	6, 5T	Android 10
Huawei	P20 Pro	Android 10
	P10	Android 9
Honor	9	Android 9
Motorola	G8 Power	Android 10

Table 1: User device/OS tested during the experiment.

3.4. Results: user (un)awareness

We report in Tables 2 and 3, respectively, the questions in the questionnaires and the most significant results, indicating the percentage of non-numeric responses and the weighted average (W/A) of numeric responses, in this case not available (N/A) is indicated in the percentage field. We emphasize that what we report here are qualitative and quantitative indicators whose objectives are: i) to show whether a particular device and its OS makes the user aware of a vulnerable configuration of the

Question	Answers	Percentage %
Does Eduroam use certificates in the authentication process?	Yes	34
	No	26
	I don't know	40
How did you configure Eduroam?	Manually	77
	Automatically with Eduroam CAT	17
	I don't use Eduroam	6
Have you ever connected to Eduroam under mobility/roaming conditions?	Yes	31
	No	60
	I tried without success	3
	I don't use Eduroam	6
How much attention will you pay to configure Wi-Fi network from 1 to 10?	Weighted average: 5.24	N/A
Does the device when it detects Eduroam, automatically connect to it?	Yes	83
	No	11
	I don't use Eduroam	6
How much would you rate your knowledge of Wi-Fi security from 0 to 10?	Weighted average: 4.74	N/A
Regarding the configuration of the certificate, choose the answer that you think more suitable	I leave the certificate field as default	31
	I have manually/automatically configured the certificate	34
	I configured "do not validate certificate"	11
	Eduroam not use certificates	17
	No answer	7
What security protocols does Eduroam support? (select all the answers you think are valid)	Generic Username/Password protocol	23
	EAP-TTLS/PAP	37
	EAP-TTLS/MSCHAPv2	11
	WPA2-Personal	23
	PEAP/MSCHAPv2	0
	I don't know	34

Table 2: Key questions from the pre-attack questionnaire.

Wi-Fi network; ii) to analyze the end users by showing their characteristics in the field of network security, highlighting the critical points; and iii) to analyze the result of the instructions provided by the organizations to the users of the Eduroam network configuration. Through the analysis of the experimental results, many important aspects were discovered that describe the current Wi-Fi network security knowledge and vulnerability awareness.

Certificate-Based Authentication: As shown in Table 2, the vast majority (66%) of the participants are unaware that the Eduroam network uses certificates in the authentication process. This is surprising as the set of users comprises mainly students in ICT disciplines and suggests that students do not behave as ICT engineers during daily activities. Combining this with the fact that 77% of the sample of students manually configure the network, it is very likely that users have vulnerable configurations that do not check the certificate's validity.

Authentication protocols: Table 2 also shows that students do not have a clear idea about the authentication protocols supported by the network: this exposes them to numerous vulnerabilities that could even lead them to reveal their IMSIs - International Mobile Subscriber Identity [1]. From the experiments, in fact, it turns out that users need to become more familiar with the Eduroam authentication protocols. During device configuration, there are hence chances that they select EAP-SIM/AKA as EAP method [35, 36] and eventually, reveal their IMSI to the attacker (and also to the legitimate Eduroam system). As the IMSI is a worldwide permanent identifier, an attacker can exploit its knowledge to carry out a series of attacks designed to locate and track a specific user.

Possible countermeasures to vulnerabilities depending on users' non-awareness necessarily includes communication campaigns to disseminate greater knowledge about authentication protocols and convince users to pay more attention when configuring the network.

3.5. Results: organization vulnerabilities

Organizations participating in the Eduroam project (in Italy there are currently 169 active Eduroam Identity Providers [37]) must provide access credentials to users. They must also provide a guide to manually configure access to the Eduroam network or provide a tool to do so automatically. Unfortunately, organizations - including the University of Rome "Tor Vergata" where we executed part of the experiments - often provide outdated and insecure guides to users, which expose them to various vulnerabilities. Since 77% of the users configure the network manually and given that our questionnaires reveal a limited (on average) understanding of Wi-Fi/802.1x security configurations, organizations play a key role in the security of Eduroam. Often, the Eduroam credentials provided to professors by their employers are the same used in national authentication and authorization services based on institutional digital identity: they can also be used for accessing several federated services, including GÈANT eduGAIN,

the inter-federation framework of Authorisation, Authentication and Identity (AAI) services that provide users across the research and education community with single-sign-on to thousands of service providers worldwide. The exposure of professors' Eduroam credentials, therefore, represents a huge issue related to the privacy and security of their digital identity. According to the latest GARR's 2020 annual report (Italian acronym for Group for the hArmonisation of Research netwoRks) [38], the number of devices that authenticated to the Italian Eduroam Federation is very high (321,000 devices of Italian users authenticated in Italy or abroad and 182,000 devices of foreign users in Italy) even if it halved compared to the previous year, probably due to travel restrictions related to the COVID-19 pandemic. As for Eduroam CAT, the number of registered Identity Providers is 81, with an overall download of profiles generated by the system reaching 1.8 million. Regarding Eduroam digital identity, federated authentication for access to services using institutional digital identity is also growing, with an average monthly number of logins exceeding 30 million accesses. Regarding the type of logins, 9 out of 10 are used to access internal services configured directly on the Identity Providers of some organization; almost 1 out of 10 is a service accessed through the IDEM Federation (i.e., the Italian Federation of Universities and Research Institutions for Authentication and Authorisation). At the same time, 1 out of 20 is a service from other federations via GÈANT eduGAIN.

Possible countermeasures to organization vulnerabilities include a continuous update of the guides and tutorials related to the network configuration and the preparation of (pre-recorded) seminars to raise users' awareness about these issues.

Question	Ans	%
You have connected to an Eduroam rogue AP, how serious do you consider this situation to be from 0 to 10?	7.09	N/A
Were there any abnormal behaviors or warnings that made you feel you were under attack? Did you realize you were under attack?	Yes	24
	No	76
How serious do you consider the leakage of credentials to be from 0 to 10?	8.85	N/A
Do you have any idea what vulnerability enabled the attack?	Yes	24
	No	76
Did you expect an Eduroam configuration that allows you to access the Internet to be insecure and allow credential leaks?	Yes	55
	No	45
After this experiment, how much attention will you pay to configure Wi-Fi network from 1 to 10?	6.94	N/A
What do you think is your level of awareness of Wi-Fi security from 0 to 10?	4.22	N/A
Do you think there is a need to raise awareness about Wi-Fi security?	Yes	97
	No	3

Table 3: Key questions from the post-attack questionnaire.

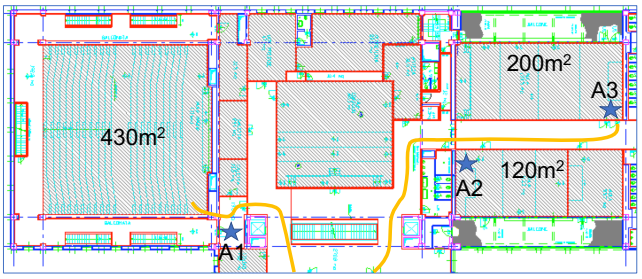

Figure 3: Rogue APs deployment in the Engineering Faculty of the University of Brescia: orange lines are paths with maximum flow of students when going to classes, having breaks, etc. A1-A2-A3 are our APs, we also indicate the surface of the three biggest classrooms of the building.

4. Experiments "in the wild"

We describe in this Section the attack that we ran "in-the-wild" at the University of Brescia. Differently from the controlled experiments in Rome, we carried out this set of tests in an unsupervised way, i.e., data were collected from a pool of entirely unaware users. By setting up rogue APs broadcasting the enterprise network named "Studenti" (Students) that is available everywhere in the campus, we restricted the analysis to students during their typical activities, i.e., attending classes, having breaks, walking in the corridors, etc. While the attack, in this case, was not against the Eduroam network, the involved mechanisms and the corresponding vulnerabilities, ranging from broken (or default) configurations of Android devices, bad practices of the users, etc, are identical. The conclusions that we can draw are hence the same as if we ran the attack against Eduroam.

To maximize the attack coverage and to verify possible biases due to the interests, attitudes or IT skills of the attacked students, we ran three distinct experimental campaigns in different areas of the campus: at the Medical School, at the Engineering Faculty, and at the Department of Economics. In each area, we deployed three APs that were active during working hours for two consecutive weeks for a total of ten capture days. We placed the three APs close to the biggest classrooms and along pathways to maximize the number of observed devices. We also configured the operating frequencies of the APs to overlap with neighboring legitimate APs in order to capture every user device when coming to their proximity. In Fig. 3, we show the resulting environment at the engineering faculty, the only one which we report for brevity: for the other two locations, we followed the same guidelines. We can see the main pathways (orange lines) followed by students to attend lectures in the three biggest classrooms in the building: the chance students' devices meet one of the three rogue APs (A1, A2 and A3) is very high thanks to the chosen positions.

We note that the attack setup was discussed and planned with the ICT manager that administers the campus network and, as we will explain later, the experiment was aimed at improving

the awareness of the problem among attacked users.

While the software and hardware tools that we adopted in Brescia are similar to those we already explained in Section 3, we provide some additional details in the following subsections as they are very specific to the new scenario. Also, the data that we captured here is much richer than in Rome, and for this reason, we start by explaining what we collected.

4.1. Data collected

Here we describe which pieces of information we intended to obtain from the experiment. These data were then used to compute various statistics for each department that could let us make comparisons between them. These are the main data we gathered:

- **1. Authentication runs.** An authentication run happens when a device senses the Wi-Fi network and tries to connect to it, regardless of the outcome. We only counted the first access attempt from each device, using MAC addresses to distinguish between devices already known to the network and new ones;
- **2. Successful runs.** An authentication run is considered successful if the device proceeds with phase 2 authentication and completes it. At this point, the device uses the rogue network as if it was the original one. As for the previous statistic, only the first successful authentication from each device was counted, where MAC addresses were used to determine whether the device did already connect to the network or not;
- **3. Anonymous identities.** We were interested in knowing the adoption rate of anonymous identities among users in order to understand how many of them exchange their personal, unique identities in the clear instead of an anonymous one, thus compromising their anonymity. The anonymous identity generally advised is "anonymous", however, some organizations impose restrictions on this, thus forcing users to use their personal identity; this is not the case of the University of Brescia, but some users that have learned about the anonymous identity might have changed this field in the specific configuration of the "Studenti" network;
- **4. MAC address randomization.** MAC address randomization is now implemented and enabled by default on virtually any recent smart device. Generally, MAC address randomization guarantees anonymity to the user when connecting to different networks, as the randomized MAC address remains typically the same when connecting to Wi-Fi networks with the same SSID;
- **5. Phase 1 authentication**. We set up our rogue RADIUS server to support only EAP-TTLS and EAP-PEAP methods for phase 1 authentication. Nevertheless, we wanted to discover how many devices were configured (possibly) manually by their owners to use one or the other. Security-wise, the two methods are virtually identical;
- **6. Phase 2 authentication.** This step of the authentication is the most crucial as if a weak method is agreed upon or chosen by the supplicant, then the attack can retrieve the password. In our case, the rogue RADIUS server was set to always propose EAP-GTC to the supplicant so that the password is delivered as plaintext if accepted by the supplicant. Many other authentication methods were supported to let more users authenticate to the rogue RADIUS server;
- **7. Apple devices**. After the experiment in Rome, we were expecting mainly Android devices would fall in the attacks. Still, in this unsupervised experiment, users interacting with their Apple devices might manually force the supplicant into validating the rogue certificate and connect, as introduced in Section 3.3. To distinguish such cases, we analyzed the first DNS query made by every device to detect if it was directed to the FQDN **captive.apple.com**, which is the default behavior of Apple devices (including laptops) to understand if a further authentication via the captive portal is required to access the internet.

For reasons concerning users' privacy and data regulation, anonymous identities and inner identities were saved in a hashed form, using SHA256 as hash function. Because of this, the adoption of anonymous identities was measured either by comparing the stored hashed with a known one (e.g. the hash of "anonymous" or "Anonymous"), or by checking if the hashes of the outer and the inner identity for an authentication run were the same, which confirms user did not configure an anonymous identity.

In order to comply with data regulations imposed by the ICT team of the university of Brescia, we did not store any information tied to the collected identities, not even password hashes or authentication challenges.

4.2. Hardware tools

In order to recreate a stand-alone rogue network, some hardware that could run both a RADIUS server and a wireless AP was needed. For reasons related to the feasibility of the attack in the wild, we decided to use three **Raspberry Pi model 3B+**: they are cheaper than higher-end general-purpose computers, they are smaller in size, so they can be hidden easily, and they have access to all the software needed to create a complete 802.1x network. Since the Wi-Fi Broadcom chipset embedded in the Pis cannot create WPA-Enterprise networks because of incompatibility issues in the drivers, we adopted as wireless interfaces three USB TP-Link antennas, which also provided a higher coverage than the internal interface.

We connected each Raspberry Pi to a Power-Over-Ethernet switch, specifically a **Hewlett-Packard ProCurve 2600-8-PWR**, by adding a PoE shield to the Pi motherboard. This was necessary in order to force a remote reboot of the devices: Pi devices, in fact, turned out to be unreliable, especially under conditions of massive authentication load that was leading to frequent crashes. We implemented a software solution using

the *expect* scripting language so that each Raspberry Pi could periodically check if any of the other two is down and, in that case, restart it by forcing a power cycle of the corresponding PoE port.

4.3. Software tools

In this section, we present the pieces of software we used to recreate the WPA-Enterprise "Studenti" network.

As we did in Rome, **FreeRADIUS** v3.0 was our choice for recreating the RADIUS server component. However, in order to carry out the attacks with higher flexibility than in Rome, for example, to limit the attack only to devices that are connecting for the first time and to allow as many users as it was possible to complete the authentication, we performed an intense modification of the authentication flows by modifying the "default" and "inner-tunnel" configuration files of FreeRADIUS. We developed a Python script to blacklist from hostapd devices that were successfully attacked: this feature was requested by the ICT manager to avoid involving into the experiment the same user over and over again. We added other Python and bash scripts to log data of interest and to have better control on how the authentication flow proceeded, i.e., for sending the RADIUS Access-Accept packet only at a specific step of the authentication to maximize the odds of having it accepted by the supplicant.

To recreate the wireless network, we also used here the **hostapd** application, configured similarly as in Rome to provide 802.1x authentication by connecting to the RADIUS server.

To capture the DNS queries executed by users' devices, we were starting a **tcpdump** v4.99 instance restricted to capturing UDP traffic on port 53 at every successful authentication, i.e., every time a device fell victim to the attack. This information helped in some cases to identify the brand of the device, which was also a topic of interest.

Finally, we implemented a **captive portal** that was presented to users once connected to the rogue network: we report it in Figure 4. In our intention, this should have served to let victims know that they connected to a non-legitimate network, i.e., their behavior or the (mis)configuration of their device exposed them to a possibly dangerous attack. To this end, we implemented a simple self-hosted website using the **Flask** framework and the **dnsmasq** software to redirect specific DNS queries to our local web server, thus letting the device know that the network uses a captive portal which, in our case, was purely informative.

4.4. Phases of the attack

We attacked the three locations during different and non-adjacent periods of the year 2022: the setup, in fact, was just one, and we moved it among the three different places following the dates proposed by the ICT staff. We first attacked the **Medical School**. There, the three evil twins operated from April 22nd to May 6th. The starting day corresponds to the end of the Easter break, which is dedicated to exam sessions, after which classes restart. Shortly after, we moved the equipment to the **Department of Economics** where the attack was active from May 17th to May 30th. It is to be noted that classes were over by that time:

Figure 4: Captive portal shown to victims for increasing their awareness about the threats they got exposed to because of their behavior or device misconfiguration.

this may explain the lower numbers that reduce the size of the data sample and might have some impact on the statistics. We finally attacked the **Engineering Faculty** from October 3rd to October 14th: the long pause was decided to avoid the summer break and we hence waited until the new academic year started so that classes were again full of students.

To summarise, we report in Figure 5 the daily amount of devices that started an authentication run with the evil twins (blue) and the number of devices that completed it (red), thus falling victim to the attack. Each device is only counted the first time it is authenticated by one of the rogue AP, after which it is blacklisted. This may explain why both curves in the three locations show a decreasing tendency. Curves at the Dept. of Economics confirm that the considered period of time was not optimal. Numbers, in fact, are much lower than in the other two locations. We analyze next the collected data in detail.

4.5. Analysis of the results

We try to explain in this Section the collected data by also comparing the differences between the three locations: we refer to Table 4. Besides the absolute amount of users, which varies significantly from one location to another, we see that:

- **Security awareness**. Engineering students are less prone to fall in the attack with respect to Medical School students (25% vs 39.2%). This is probably due to a higher security awareness among engineering students because of their affinity with the topic. While a similar conclusion can be preliminarily drawn for the students at the Dept. of Economics, the absolute number of successful runs is signifi-

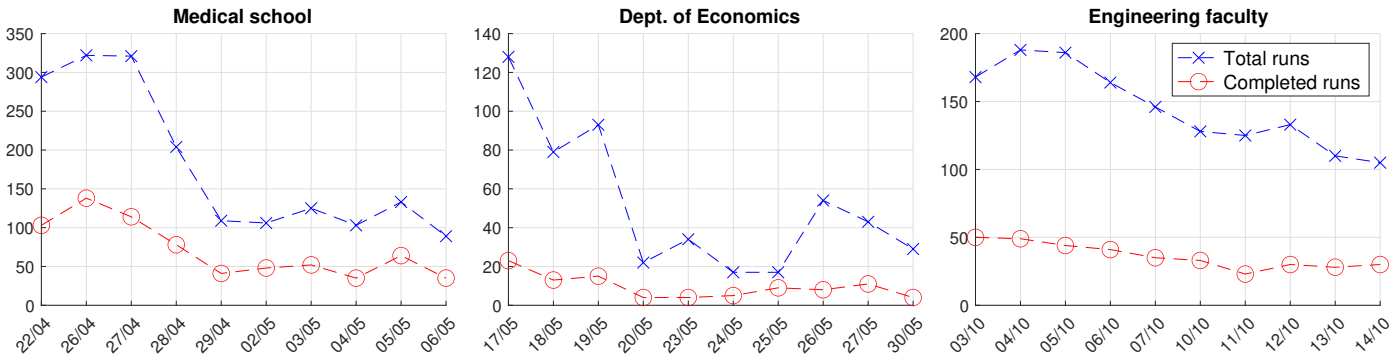

Figure 5: Daily connection attempts (blue) and attacked devices (red) of unique devices: blacklisting MAC addresses on each evil twin exempted the same victims from being attacked more than once.

	Medical school		Dept. of Economics		Engineering Faculty	
Statistic	**Value (abs)**	**Value (%)**	**Value (abs)**	**Value (%)**	**Value (abs)**	**Value (%)**
Total runs	1806		516		1453	
Successful runs	708/1806	39.2%	96/516	18.6%	363/1453	25%
Non-random MACs	752/1806	41.6%	98/516	18.9%	323/1453	22.2%
Anonymous identities	0/1806	0%	0/516	0%	0/1453	0%
Phase 1 - PEAP	708/708	100%	98/516	100%	361/363	99.5%
Phase 1 - TTLS	0/708	0%	0/516	0%	2/363	0.5%
Phase 2 - GTC	345/708	48.7%	13/96	13.5%	54/363	14.9%
Phase 2 - MS-CHAPv2	225/708	31.8%	31/96	32.3%	127/363	35%
Phase 2 - PWD	138/708	19.5%	52/96	54.2%	182/363	50.1%
Apple devices	103/708	14.6%	0/96	0%	4/363	1.1%

Table 4: Detailed statistics of the attack in the three locations.

cantly lower, 60% less than those at the Engineering faculty. This makes the corresponding success rate statistic of 18.6% less reliable in drawing definitive conclusions about the overall security awareness of students at the Dept. of Economics. In order to obtain more accurate and reliable results, it would be necessary to conduct additional experiments targeting the Dept. of Economics, increasing the sample size, and thus providing a more comprehensive understanding of the security awareness among the students in this department; still, this applies only at the Dept. of Economics leaving conclusions about the other two Depts. unchanged;

- **Guidelines**. The statistics computed on the phase 2 authentication methods suggest that users from all three departments are not following the network admins' guidelines when configuring the network on their devices. As shown in table 4, the most widely used authentication method is not MS-CHAPv2 in any of the departments, despite it being the suggested one in the university's official guide;

- **Apple devices**. Apple devices can fall into the attack only if users force them because of bad practices. This behavior has been mostly observed at the Medical School, where 103 different Apple devices completed the authentication against the evil twin. This, in combination with the fact that all of them agreed on using GTC (as proposed by the rogue RADIUS server) severely degraded the security position of those users as their password was captured as plaintext.

- **Phase-2 method**. Modern Android devices generally have a phase-2 authentication method set to either MS-CHAPv2 or PWD by default. Older Android devices and all iOS devices, instead, do not have a phase-2 authentication method explicitly set, thus they agree on what is proposed by the RADIUS server (if, of course, supported by the supplicant too). However, it seems that there are still many devices that do not have the phase-2 method set. As can be seen, among the three locations, 412 devices (of which 107 were Apple ones) authenticated against the rogue RADIUS server with GTC. This is a major security concern as an attacker could have gained instant access to 412 plaintext passwords, thus compromising a high number of accounts with slim resources;

- **Anonymous identities**. Anonymous identities are not used at all by users in any of the three locations. This means that every user is open to sharing her/his personal identity (which is strongly tied to the user) with anyone;

- **MAC randomization**. MAC randomization, albeit being implemented and active by default on virtually any recent

device, is still not that common. As it can be noted, in the Medical School 41.5% of the devices did not use a randomized MAC address. At the Engineering Faculty and Dept. of Economics we find much lower rates, respectively 22.2% and 19%.

Finally, we note that the data collected "in-the-wild" on the campus of Brescia confirms what was already observed during the controlled experiments in Rome: the 802.1x protocol applied to the authentication of users in a Wi-Fi network is not adequate to guarantee their security. Too many users, when their devices do not automatically connect, force the supplicant into connecting without caring about any security advice. Simply said, the "connect-at-your-own-risk" option should not be proposed to users at all: the level of awareness about security threats, according to what we observed, is extremely low also in those communities of users (like students at the Engineering Faculty) that may have also received some classes on the topic. As evidence, we note that not even a single message was received on the email box advertised on the captive portal, where instead we were expecting users to ask for clarification, if not on what happened, about who had the ability to read their personal data: we can easily conclude that users do not care at all.

The final solution to counter this issue would be designing a new EAP method that does not use the user password. For instance, it may mutually authenticate the device to the network by using cryptographic long-term keys derived once through a specific application running on the device when it is connected through other technologies (LTE/5G, already trusted Wi-Fi) starting from the password. However, as this is a long-term solution, at the moment all manufacturers should simply deactivate the possibility of forcing the supplicant when the certificate changes. We found the first example in this direction in iOS 16, as we describe in the next Section.

5. User device behavior analysis

We present in this section an in-depth behavioral preliminary analysis of four smartphones from different brands: Apple, Samsung, Xiaomi, and ZTE. We evaluate the feasibility of the attack over different versions of the OS, of the devices' network configuration, and of the type of certificate used in the malicious network (self-signed, expired and valid). In addition to describing the differences in behavior and highlighting the vulnerabilities, we also propose solutions to the discovered security problems.

5.1. Attack feasibility

We start by evaluating the influence of the chosen authentication protocol and the type of certificate used.

Authentication protocol: By analyzing the results shown in Table 5, it is possible to show that only one smartphone has an implementation of Android OS that allows extremely high attack feasibility because any configuration that allows access to the legitimate Eduroam network is vulnerable and allows attackers to obtain access credentials of the victim under almost all tested conditions. Evaluating table 5 from the point of view

Device and OS	**TTLS-PAP**			**TTLS/PEAP-MSCHAPv2**		
	Self signed	Exp.	Valid	Self signed	Exp.	Valid
Apple iPhone 11, 12, 13, 14 Apple iPad Pro 13" iOS 16	A	A	A	N/A	N/A	N/A
Apple iPhone 11, 12, 13 Apple iPad Pro 13" iOS 15	A	A	A	N/A	N/A	N/A
Apple iPhone XS iOS 14.4	A	A	A	N/A	N/A	N/A
Google Pixel 6 Pro Android 13	A	A	A	N/A	N/A	N/A
Xiaomi Mi 10 Android 11	C	C	D	A	A	A
ZTE Axon 10 Pro Android 9	B	B	B	A	A	A
Samsung S8 Android 8	B	B	B	A	A	A

Table 5: Attack feasibility for different devices under different conditions: A : Good, B : Not so good, C : Not so bad, D : Bad.

of the feasibility of the attack, the letters shown have the following meanings:

- **A)** Indicates that the device does not connect to the rogue AP with any certificate verification policy configuration, consequently the feasibility of the attack is null.
- **B)** Means that the device connects to the rogue AP only with the "Do not validate" as certificate control policy.
- **C)** Indicates that the device connects to the malicious AP with both the "Do not validate" and "Please select" policies, while using the "Use system certificate" policy the phone does not connect to the rogue AP.
- **D)** Emphasises a vulnerable behavior that makes the attack feasible also with the "Use system certificate" control policy.

In addition to the protocol shown in Table 5, the configuration using Eduroam CAT is also analyzed under which in all possible cases, the behavior belongs to the letter **A)** except with the Google Pixel 6 Pro with Android 13, which fails authentication with the Eduroam network probably because the validation process of the server side certificate failed. It only allows connection with the Eduroam network by specifying the "Use system Certificate" certificate control policy and, consequently, the sub-policy "Do not validate" and "Request Certification Status". Interesting are also the cases in which the user chooses either the 'Require Certificate Status' sub-policy or the TOFU policy for which the phone fails to connect to the Eduroam network, and furthermore, in the second case, the user is notified that the 'chain is invalid' indicating that the certificate verification during the authentication process fails.

Certificate: Experiments show that the type of certificate used in the malicious network only affects one smartphone. Unlike other tested smartphones, if the rogue AP uses a valid cer-

tificate, it provides user credentials even if the user explicitly selects the "Use system certificate" policy. This leads us to believe that there may be vulnerabilities in the Wi-Fi network configuration and connection software. In all cases where the user connects to the malicious network, the device provides plain text credentials to the attacker without the user's awareness. The tested Eduroam Wi-Fi network uses the Wi-Fi Alliance WPA2 standard while the new WPA3 [39] standard is available. WPA3 Enterprise has improved security compared to WPA2. This is largely due to the inability to use "skip certificate verification" or "accept any certificate" to configure the supplicant, and if the supplicant cannot verify the server identity, then it will no longer be possible to send user credential materials to AS automatically; it requires that the user explicitly accepts the trust in the certificate provided by AS. This relatively straightforward constraint on allowed configuration and user participation reduces the chance of tricking a supplicant into leaking credentials. It is important to point out that the WPA3-Enterprise specification does not impose any timing or causality between a failed server identity verification and the user's decision, which may still be vulnerable.

5.2. Device misconfiguration protection

By evaluating the results shown in Table 5 from the perspective of Wi-Fi network misconfiguration protection:

- **(A)** and **(B)** can be considered reasonable: in the case **(B)**, the user can, in fact, misconfigure the phone but only via a deliberate action.
- **(C)** and **(D)** can be considered dangerous: we consider dangerous not only **(D)** (for obvious reasons) but also **(C)**, since in this case, the default configuration is vulnerable, whereas the vulnerability is fixed only by an explicit intervention of the end user which must set the "Use system certificate" policy.

Moreover, results show that, despite the security improvements introduced in Android 7.0 and Android 13 described in detail in section 3.3, one among the four in-depth tested smartphones did not implement these enhancements. For the sake of clarity, however, after the latest update released by the manufacturer at the beginning of 2022, the device no longer appears to be vulnerable. Regarding Apple's devices, as mentioned earlier, Apple's policy of reducing as much as possible the configuration capabilities provided to end users appears to pay off in terms of security, as it was not possible for any user to misconfigure the device and expose it to the attacks tested in this paper. Furthermore, Apple smartphones always show the certificate to the user when connecting to the network and whenever a new one is provided, therefore raising the user's level of awareness. In addition, Apple smartphones always show the certificate to the user when connecting to the network, and each time a new one is provided, thus increasing the level of user awareness. However, it is important to emphasize that iOS 16 appears not to implement certificate validation correctly; in fact, while deliberately accepting the new certificate provided by the network, the connection fails, and the user is notified that he cannot connect. While this seems to be a sort of bug (why asking to trust the new certificate if the choice is not registered), its final effect is to improve the security.

5.3. Correlation between user awareness and phone model

Our findings indicate that user awareness plays a significant role in mitigating the risks associated with device misconfiguration. For instance, Apple devices have consistently demonstrated higher levels of user awareness due to their design and approach to presenting information to the end user. Apple smartphones always display the certificate when connecting to a network and each time a new one is provided, effectively raising user awareness levels. This approach has proven effective in reducing the likelihood of users misconfiguring their devices and exposing them to potential attacks. On the other hand, Android devices have shown varying levels of user awareness depending on the specific phone model and manufacturer. Some Android devices have implemented security improvements introduced in Android 7.0 and Android 13, while others have not. This inconsistency can lead to varying levels of user awareness and potentially higher risks for users with devices that have not implemented these security enhancements. It is important to note that increasing user awareness can significantly contribute to reducing the risks associated with device misconfiguration. Educating users about the potential dangers and providing clear instructions on how to configure their devices properly can be an effective approach to improving overall security. In conclusion, there is a clear correlation between user awareness and phone model, with higher levels of user awareness leading to better security outcomes and reduced risks associated with device misconfiguration.

6. Related work

Regarding 802.1X authentication protocols and Wi-Fi enterprise security assessment, many scientific publications related to our work [40, 41, 42, 43, 44, 31, 45, 46] highlight that Wi-Fi enterprise networks are vulnerable to different attacks in several aspects, also using specific adversary models (i.e, enterprise matrix MITRE ATT&CK for corporate networks [47]) and propose possible countermeasures to such vulnerabilities [31]. MitM attack design and implementation were discussed in all papers. In [40], an exhaustive security analysis of WPA2 enterprise network configurations is carried out, and a framework for comparison is proposed. In [41, 31, 45, 46], the main objectives are (i) to study the feasibility of the attack (trying to maximize it, for instance, through the use of external jammers [45]) by analyzing the data exfiltrated by the attacker on different target operating systems (or on a specific one, i.e., iOS [46]); and (ii) assess the impact of the attack by exploiting this data (i.e., association maps between Eduroam users and people [41]). An analysis more addressed to the assessment of the certificate validation procedure was carried out in [44], while a 'different' and more related look at the world of mobile networks is shown in [43], where it is shown how to exploit the

EAP-SIM/AKA authentication protocols of Enterprise Wi-Fi networks to implement IMSI Catching attacks. Although Wi-Fi enterprise security assessment has been extensively explored in the works mentioned above, to the best of our knowledge, our work is the most extensive study to date that also covers less technical aspects related to user awareness by exploiting questionnaires as social research tools to analyze this 'worrying' social phenomenon.

7. Conclusion

The main contribution of this paper is an experimental investigation (i) on the practical security of the 802.1x mechanisms designed for the authentication of users in Wi-Fi enterprise networks in general, and, more specifically, (ii) on the security of Eduroam, a network used by millions of users in the research and education sector in more than 100 countries worldwide. To this end, we implemented and ran attacks in two different Universities in Italy aimed at assessing vulnerabilities in the users' configurations in order to steal their credentials. Moreover, an in-depth analysis was conducted on some specific smartphones, to understand their behavior better while varying the authentication protocols and the nature of the certificates employed (self-signed vs. expired vs. valid). Finally, the technical work was complemented in one campus by the proposal of two anonymous questionnaires (one before the attack, and one after it) to the volunteers, so that they can gather insights on their security awareness; and in the other campus with the presentation on the mobile device of victims (through a specific captive portal) of a warning message reporting what happened, again aimed at improving users' awareness about security threats.

A technical lesson learned from our work is that Android, compared to iOS, offers greater freedom of configuration to the user and therefore is more exposed to possible attacks - like the one implemented by us - that aim at exploiting wrong or incomplete configurations. Moreover, our results appear to show that the practical sanity of WPA-Enterprise networks is quite low. Indeed, with somewhat consolidated attack techniques, an attacker can still steal long-term credentials from a non-marginal fraction of users (more than one-third in our tests). Note that at least in the Eduroam case the attack yields the malicious gathering of persistent credentials used for a multiplicity of critical single-sign-on educational services (e.g., access to university email, access/recording of exam grades, etc.). In the case of Eduroam, this appears to be a severe vulnerability considering the highly interconnected global environment in which this network is embedded. Indeed, our experiments show that this overall lack of security in many cases could be easily avoided with a few straightforward patches to the supplicant only in order to properly use the anonymous identity, rejecting some weak authentication methods like GTC, or adopting the MAC address anonymization policy beyond the phase of network scanning.

Another lesson is that we would have expected a greater security awareness in the set of users involved in the experiments - young university students, some of which attending ICT-related programs - as well as in the organizations that are called to configure the network access properly. Despite the fact that the considered vulnerabilities have been documented in the literature, we found a substantial discrepancy with the practical security awareness level of either users as well as organizations - even if we conducted tests in our institutions, we are aware of many other institutions which suggest to their students and employees vulnerable configurations. Even more shocking, however, is the users' disregard for their own security: the absence of emails after being redirected to the captive portal and the fact that many users consciously forced their (initially resilient) device to fall victim to an attack jointly represent an indication of disregard for the consequences of such choices; this could be partially justified because the context of the attack, thus the university environment, is not perceived by users as an area where sensitive data can be compromised.

As future work, we plan to shift the focus from the implementation and evaluation of the attack itself to the analysis, design, and development of technologies that can: (i) assist end users and organizations in detecting possible attackers; and (ii) raise awareness of users' bad practice by adding to legitimate APs secondary Virtual rogue ones that keep communicating something went wrong, i.e., by using the captive portal approach we demoed in this work, by sending targeted emails or advertising, etc. For this second task, a deep social analysis is required to avoid creating false alarms, spreading panic, or even making the entire effort useless because of its repetitiveness.

References

[1] T. Perković, A. Dagelić, M. Bugarić, and M. Cagalj. On wpa2-enterprise privacy in high education and science. Security and Communication Networks, 2020, Sept. 2020.

[2] Christopher P. Kohlios and Thaier Hayajneh. A comprehensive attack flow model and security analysis for wi-fi and wpa3. Electronics, 7(11), 2018.

[3] A. Bartoli, E. Medvet, A. De Lorenzo, and F. Tarlao. (in) secure configuration practices of wpa2 enterprise supplicants. In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES), 2018.

[4] M Vanhoef. Windows 10 lock screen: abusing the network ui for backdoors (and how to disable it). Mathy Vanhoef blog, 2017.

[5] Yulong Zou, Jia Zhu, Xianbin Wang, and Lajos Hanzo. A survey on wireless security: Technical challenges, recent advances, and future trends. Proceedings of the IEEE, 104(9):1727–1765, 2016.

[6] V. Ramachandran. Cracking wpa/wpa2 personal and enterprise for fun and profit. In Hacktivity Conference, 2012.

[7] Raheem Beyah and Aravind Venkataraman. Rogue-access-point detection: Challenges, solutions, and future directions. IEEE Security and Privacy, 9(5):56–61, 2011.

[8] Stuart Minshull and David Starobinski. An empirical study of wpa-enterprise misconfigurations.

[9] CENSUS. The known beacons attack (34th chaos communication congress). `https://census-labs.com/news/2018/02/01/known-beacons-attack-34c3/`, Last accessed on 2022/11/09.

[10] G Chatzisofroniou. Lure10: Exploiting windows automatic wireless association algorithm, 2017.

[11] GEORGE Chatzisofroniou. Efficient wi-fi phishing attacks. 2016.

[12] AirTight Networks. Hole196 vulnerability. `http://securedsolutions.com.my/pdf/WhitePapers/WPA2-Hole196-Vulnerability.pdf`, Last accessed on 2022/11/09.

[13] SensePost. Sensepost: Manna from heaven. def con 22 (2015). `https://sensepost.com/blog/2015/improvements-in-rogue-ap-attacks-mana-1/2/`, Last accessed on 2022/11/09.

[14] D.A. Dai Zovi and S.A. Macaulay. Attacking automatic wireless network selection. In Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, pages 365–372, 2005.

[15] Nadarajah Asokan, Valtteri Niemi, and Kaisa Nyberg. Man-in-the-middle in tunnelled authentication protocols. In International Workshop on Security Protocols, pages 28–41. Springer, 2003.

[16] Mathy Vanhoef and Eyal Ronen. Dragonblood: Analyzing the dragonfly handshake of wpa3 and eap-pwd. In 2020 IEEE Symposium on Security and Privacy (SP), pages 517–533, 2020.

[17] Rajeev Singh and Teek Parval Sharma. On the ieee 802.11 i security: a denial-of-service perspective. Security and Communication Networks, 8(7):1378–1407, 2015.

[18] Mallikarjun Hangargi. Denial of service attacks in wireless networks. Northeastern University, 2015.

[19] Jens Steube. New attack on wpa/wpa2 using pmkid. Hashcat: website, 2018.

[20] J. Snodgrass and J. Hoover. Byo-disaster and why corporate wireless security still sucks, 2013.

[21] Mathy Vanhoef and Frank Piessens. Key reinstallation attacks: Forcing nonce reuse in wpa2. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, page 1313–1328, New York, NY, USA, 2017. Association for Computing Machinery.

[22] Mathy Vanhoef and Frank Piessens. All your biases belong to us: Breaking rc4 in wpa-tkip and tls. In Proceedings of the 24th USENIX Conference on Security Symposium, SEC'15, page 97–112, USA, 2015. USENIX Association.

[23] K. Wierenga, S. Winter, and T. Wolniewicz. The eduroam architecture for network roaming. RFC 7593, 2015.

[24] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible authentication protocol (eap). RFC 3748, 2004.

[25] D. Simon, B. Aboba, and R. Hurst. The eap-tls authentication protocol. RFC 5216, 2008.

[26] P. Funk and S. Blake-Wilson. Extensible authentication protocol tunneled transport layer security authenticated protocol version 0 (eap-ttlsv0). RFC 5281, 2008.

[27] Microsoft. MS-PEAP: Protected extensible authentication protocol (peap), 2018. `https://msdn.microsoft.com/en-us/library/cc238354.aspx`, Last accessed on 2021/5/26.

[28] Pragati Shrivastava, Mohd Saalim Jamal, and Kotaro Kataoka. Evilscout: Detection and mitigation of evil twin attack in sdn enabled wifi. IEEE Transactions on Network and Service Management, 17(1):89–102, 2020.

[29] Zhanyong Tang, Yujie Zhao, Lei Yang, Shengde Qi, Dingyi Fang, Xiaojiang Chen, Xiaoqing Gong, and Zheng Wang. Exploiting wireless received signal strength indicators to detect evil-twin attacks in smart homes. Mobile Information Systems, 2017, 2017.

[30] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Stefanos Gritzalis. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials, 18(1):184–208, 2015.

[31] Sebastian Brenza, Andre Pawlowski, and Christina Pöpper. A practical investigation of identity theft vulnerabilities in eduroam. In Proceedings of the 8th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '15, New York, NY, USA, 2015. Association for Computing Machinery.

[32] S. Tan. Display warning if users does not provide ca cert in eap config, 2016. `https://android.googlesource.com/platform/packages/apps/Settings/+/03a117b`, Last accessed on 2021/5/26.

[33] S. Tan. Add menu options for not specifying a eap ca cert and user cert, 2016. `https://android.googlesource.com/platform/packages/apps/Settings/+/f827c92`, Last accessed on 2021/5/26.

[34] Google Android. Trust on first use (tofu), 2022. `https://source.android.com/docs/core/connect/wifi-tofu`, Last accessed on 2022/11/10.

[35] H. Haverinen and J. Salowey. Extensible authentication protocol method for global system for mobile communications (gsm) subscriber identity modules (eap-sim). RFC 4186, 2006.

[36] J. Arkko and H. Haverinen. Extensible authentication protocol method for 3rd generation authentication and key agreement (eap-aka). RFC 4187, 2006.

[37] GARR. Eduroam member institutions. `https://www.servizi.garr.it/en/eduroam/users/member-institution`.

[38] GARR. Annual report 2020. `https://www.garr.it/it/chi-siamo/documenti/annual-report/5563-annual-report-2020/file`.

[39] Wi-Fi Alliance. Wpa3 specification 3.0. `https://www.wi-fi.org/file/wpa3-specification`, Last accessed on 2021/5/26.

[40] Man Hong Hue, Joyanta Debnath, Kin Man Leung, Li Li, Mohsen Minaei, M. Hammad Mazhar, Kailiang Xian, Endadul Hoque, Omar Chowdhury, and Sze Yiu Chau. All your credentials are belong to us: On insecure wpa2-enterprise configurations. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, CCS '21, page 1100–1117, New York, NY, USA, 2021. Association for Computing Machinery.

[41] B. Altinok. eduroam: Collect, track, hack. `https://medium.com/@besimaltnok/eduroam-collect-track-hack-183e843f7efc`, Last accessed on 2021/5/26.

[42] A. Bartoli, E. Medvet, and F. Onesti. Evil twins and wpa2 enterprise: A coming security disaster? Computers and Security, 74:1–11, 2018.

[43] Piers O'Hanlon and Ravishankar Borgaonkar. Wifi-based imsi catcher. In Proccedings of the Black Hat Europe 2016 Conference, London, 3rd November, volume 2016, 2016.

[44] Chad Brubaker, Suman Jana, Baishakhi Ray, Sarfraz Khurshid, and Vitaly Shmatikov. Using frankencerts for automated adversarial testing of certificate validation in ssl/tls implementations. In 2014 IEEE Symposium on Security and Privacy, pages 114–129, 2014.

[45] Aldo Cassola, William K Robertson, Engin Kirda, and Guevara Noubir. A practical, targeted, and stealthy attack against wpa enterprise authentication. In NDSS, 2013.

[46] Pieter Robyns, Bram Bonné, Peter Quax, and Wim Lamotte. Short paper: Exploiting wpa2-enterprise vendor implementation weaknesses through challenge response oracles. In Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '14, page 189–194, New York, NY, USA, 2014. Association for Computing Machinery.

[47] Maxim Kovtsur, Andrey Minyaev, Dmitrii Khramtsov, and Georgii Abramenko. Investigation of attacks and methods of protection of wireless networks during authorization using the ieee 802.1x protocol. In The 5th International Conference on Future Networks and Distributed Systems, ICFNDS 2021, page 555–561, New York, NY, USA, 2022. Association for Computing Machinery.