SSID Confusion: Making Wi-Fi Clients Connect to the Wrong Network

Héloïse Gollier
DistriNet, KU Leuven
Leuven, Belgium
heloise.gollier@kuleuven.be

Mathy Vanhoef
DistriNet, KU Leuven
Leuven, Belgium
Mathy.Vanhoef@kuleuven.be

ABSTRACT

When using protected Wi-Fi protocols such as WPA2 and WPA3, the access point that you connect to is authenticated by the client. This prevents an adversary from creating a rogue clone of the Wi-Fi network, and implies that the name of a network, called SSID, cannot be spoofed. However, in this paper we demonstrate that a client can be tricked into connecting to a different protected Wi-Fi network than the one it intended to connect to. That is, the client's user interface will show a different SSID than the one of the actual network it is connected to. The root cause is a design flaw in the IEEE 802.11 standard, causing the SSID to not always be authenticated. We demonstrate the practical impact of this attack, find that all tested devices are vulnerable to the attack, and propose backwards-compatible defenses as well as updates to the standard.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; • **Networks →** *Protocol testing and verification.*

KEYWORDS

Rogue AP, evil twin, authentication, Wi-Fi Protected Access

ACM Reference Format:
Héloïse Gollier and Mathy Vanhoef. 2024. SSID Confusion: Making Wi-Fi Clients Connect to the Wrong Network. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '24), May 27–30, 2024, Seoul, Republic of Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3643833.3656126

1 INTRODUCTION

When connecting to a protected Wi-Fi network, all transmitted data will be encrypted and authenticated. One would also expect that the name of the Wi-Fi network as shown by the client, called the Service Set Identifier (SSID), is trustworthy. In other words, if a client believes, and shows to the user, that it is connected to the protected Wi-Fi network `TrustedNet`, then one would expect that an adversary cannot trick the client into showing a different SSID.

We start by showing that significant trust is placed in the SSID that a client is connected to. For instance, many VPNs, such as Clouldflare's WARP, hide.me, and Windscribe, can automatically disable the VPN when connected to a trusted Wi-Fi network. These apps recognize the Wi-Fi network based on its SSID.

In our attack, when the victim wants to connect to the network `TrustedNet`, we trick it into connecting to a different network `WrongNet` that uses similar credentials. As a result, the victim's client will think, and show the user, that it is connected to `TrustedNet`, while in reality it is connected to `WrongNet`. The root cause is that, although passwords or other credentials are mutually verified when connecting to a protected Wi-Fi network, the name of the network is *not* guaranteed to be authenticated. This is caused by a flaw in the 802.11 standard that underpins Wi-Fi.

A common attack scenario is when networks use different SSIDs, but the same credentials, for each frequency band, e.g., for the 2.4 and 5 GHz bands. Often the 5 GHz band is preferred by clients and better secured [10]. However, our attack can *downgrade* clients to the less secure 2.4 GHz SSID. Furthermore, we demonstrate how our attack may cause a victim to automatically turn off its VPN and possibly allow the interception of the victim's traffic. The vulnerability was assigned CVE-2023-52424.

Finally, we propose three possible mitigations against our attack: a modified version of beacon protection, avoiding credential reuse, and authenticating the network SSID.

To summarize, our contributions are:

- We propose new threat models that highlight the importance of authenticating a Wi-Fi network's SSID (Section 2).
- We introduce the SSID confusion attack and systematically inspect all Wi-Fi authentication methods to determine whether they are vulnerable (Section 3).
- We evaluate our attack against various clients and networks and test an optimized variant of our attack (Section 4). Our code is available online.[1]
- We propose defenses against our attack (Section 5).

Finally, we give an overview of related work in Section 6, and we conclude in Section 7.

2 BACKGROUND AND MOTIVATION

In this section, we introduce relevant authentication methods defined in the IEEE 802.11 standard that underpins Wi-Fi [1].

2.1 Network detection and connection

Joining a network starts with network discovery. Each access point periodically sends out beacon frames containing information about the network, including its SSID. Clients can find networks around them by capturing these frames. Alternatively, a client can actively

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WiSec '24, May 27–30, 2024, Seoul, Republic of Korea

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0582-3/24/05…$15.00
https://doi.org/10.1145/3643833.3656126

[1] https://github.com/vanhoefm/ssid-confusion-hostap

look for a network by sending out a probe request. This frame contains information about the client, and can optionally contain a specific SSID. A network that hears a probe request containing its SSID should respond with a probe response containing information about its capabilities. All APs should respond to probe requests that do not contain a specific SSID.

Probe requests and responses are never authenticated. However, starting from Wi-Fi 7, access points must support beacon protection. When beacon protection is enabled, clients can verify the integrity of received beacons after they have connected to the network [15].

2.2 Home and Enterprise Authentication

There are two types of protected Wi-Fi networks: home and Enterprise networks. Home networks are protected by a pre-shared password that all users possess. In contrast, Enterprise networks use the 802.1X protocol for authentication. This enables the network to use any Extensible Authentication Protocol (EAP) it desires: authentication can be done based on a username and password, using certificates, one-time passwords, and so on.

Wired Equivalent Privacy (WEP) was the first protocol introduced to secure the communication between client stations and Access Points (APs) and was mainly aimed towards home networks. The clients and access points all share a secret key that they use to encrypt their traffic using the RC4 cipher.

Nowadays, modern Wi-Fi networks rely on a 4-way handshake to authenticate themselves and the clients, as well as to negotiate keys to encrypt the connection. The 4-way handshake takes a shared Pairwise Master Key (PMK), which can be derived differently depending on the version of Wi-Fi and the specific authentication protocol being used.

For WPA1/2 home networks, the PMK is derived using a hash of the password, the SSID, and the length of the SSID. For WPA3, an SAE handshake derives the PMK by converting a pre-shared key into a group element P, and then combining P with nonces from the client and AP. The group element P can be derived in two different ways, which we will refer to as SAE-loop and SAE-const.

With SAE-loop, P is derived using the pre-shared key and the MAC addresses of both the client and the access point. A while loop is executed until a valid point P is found on an elliptic curve. In contrast, with SAE-const, the point P is calculated in constant time, which is more secure. This method takes the SSID, pre-shared key and MAC addresses of the client and AP as an input.

For Enterprise networks, the derivation of the PMK is specified by the EAP protocol used by the network. One of the most popular EAP protocols in use today is PEAP-MS-CHAPv2, where the network authenticates itself with a TLS handshake using a certificate, and the client authenticates itself using a password. The TLS handshake provides a PMK. EAP-PWD is another example of an EAP protocol. The client and server authenticate each other with a password using a slight variation of the SAE-loop handshake.

2.3 Mesh and Ad Hoc Networks

A Wi-Fi network can operate in multiple modes, with two of the most common modes being infrastructure and mesh mode. In an infrastructure network, there is a single centralized Access Point (AP), and all clients connect and authenticate with this AP. In contrast, in a mesh network there is no central fixed node, and clients authenticate each other. Finally, an ad hoc network is similar to a mesh network, but in an ad hoc network, clients must be in radio range to exchange packets with each other.

In mesh networks, peers typically use SAE to establish a shared PMK. It is also possible to use 802.1X authentication in a mesh network, but this may create a single point of failure [8]. Once there is a shared PMK, the Authenticated Mesh Peering Exchange (AMPE) protocol is used to create session keys for data exchange. The AMPE handshake can be considered the equivalent of the 4-way handshake, but for mesh networks. Although the AMPE handshake can exchange data and integrity group keys, it does not explicitly support the exchange of beacon protection group key(s) [1].

2.4 Authentication Method Reuse

In practice, it can often occur that a client can use the same authentication method, and corresponding credentials, to connect to different networks, i.e., networks with different SSIDs. A common example is when different SSIDs are advertised for each frequency band, e.g., there might be two networks called `eduroam` and `eduroam-2.4` for the 5 and 2.4 GHz band, respectively. Networks in certain frequency bands typically support more features than others, as we discuss in section 3.2.1.

Another use-case is when university employees can use the same username and password to connect to both their own university network and any eduroam network.

3 THE SSID CONFUSION ATTACK

In this section, we introduce our threat model, the SSID confusion attack, and systematically analyze which authentication methods are vulnerable. The vulnerability was assigned CVE-2023-52424.

3.1 Threat Model

In our SSID confusion attack, we assume that the victim wants to connect to the network `TrustedNet`. The goal of the adversary is to make the victim connect to `WrongNet` instead. We also assume that the same credentials can be used to connect to both networks, for instance, the same enterprise credentials may work for both a trusted local university Wi-Fi network and an untrusted eduroam Wi-Fi network (see Section 4.3). Note that we do not assume that the victim has ever connected to `WrongNet` before, and more generally, the victim does not need to have `WrongNet` stored in its list of known networks. We also assume that the attacker does not know the victim's credentials, just that they use the same credentials for both `WrongNet` and `TrustedNet`.

3.2 Motivation

This section gives examples of the impact that tricking a victim into connecting to a different Wi-Fi network can have in practice.

3.2.1 Different SSID per Frequency Band.

Traditionally, Wi-Fi communication was done on a 2.4 GHz radio frequency, but a 5 GHz frequency is getting increasingly more common as it is faster. However, not all user devices support 5 GHz, which is why it is not uncommon for Wi-Fi access points to host two networks: one on the 2.4 GHz band and one on the 5 GHz band. Those networks

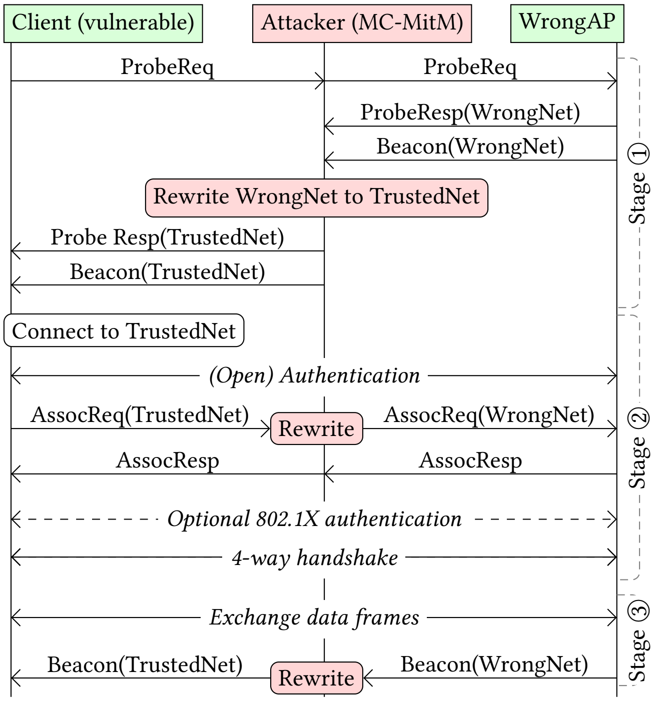

Figure 1: SSID confusion attack: the client thinks it is connecting to `TrustedNet` but in reality, it is connecting to `WrongNet`.

have different SSIDs, but use similar credentials. One issue is that APs in the 2.4 band typically support fewer security features such as management frame protection, beacon protection, or operating channel validation [10]. Additionally, APs in the 2.4 GHz band may be older [10], and therefore (still) be vulnerable to known attacks such as KRACK [17] or FragAttacks [14].

3.2.2 Auto-Disabling VPNs on Trusted Networks. Multiple VPN services, such as CloudFlare's 1.1.1.1 can be configured to turn off the VPN when connected to a trusted network. This functionality lets users type in any SSID that they trust, so the VPN is automatically turned off when the application detects that the user's device is connected to a network with that trusted SSID.

3.3 Attack Details

The attack is shown in Figure 1. First, the adversary creates a rogue AP on a channel different from a network `WrongNet`. Here `WrongNet` represents the network that we want to trick the victim into connecting with, which is advertised by `WrongAP`. The rogue AP is used to establish a multi-channel machine-in-the-middle position (MC-MitM) between the victim and `WrongNet`. In this position, the adversary forwards all frames between the victim and AP [13, 16]. This MC-MitM is feasible against all existing Wi-Fi implementations. We remark that operating channel validation can detect an MC-MitM position. However, this defense is not yet adopted in practice, and establishing a MitM would remain possible as long as the client and AP are not within radio range. All combined, establishing a MitM is a realistic assumption.

Table 1: Overview of authentication methods and whether their specification is vulnerable to SSID confusion attacks.

Network type	Authentication method	Affected
Home	WEP	Yes
	WPA1/2	No
	WPA3 SAE-loop	Yes
	WPA3 SAE-const	No
Enterprise	802.1X / EAP	Yes
Mesh	AMPE	Yes
Other	FT	No
	FILS	Yes

In stage ① of the attack, the adversary will forward any probe requests to `WrongAP`. If the probe requests contain an optional SSID equal to `TrustedNet`, then the adversary will replace `TrustedNet` with `WrongNet`, and will then forward the probe request. Similarly, any probe responses and beacons sent by the AP will be modified by replacing `WrongNet` with `TrustedNet`. As a result, the victim will think that the network `TrustedNet` is nearby, even when it is not.

In stage ② of the attack, the victim will attempt to connect with `TrustedNet`. The connection process starts by sending and receiving an (open) authentication frame that is forwarded to and from `WrongAP` without modification. Once (open) authentication is completed, the client will send an association request that includes the SSID the client is connecting to. The adversary will rewrite the SSID `TrustedNet` to `WrongNet` before forwarding the association request to the AP. The association response does not contain an SSID and can therefore be forwarded to the client without modification [1, Table 9-35]. After association, an 802.1X authentication handshake is performed when connecting to an enterprise Wi-Fi network. Finally, the 4-way handshake is used to negotiate fresh session keys to encrypt and authenticate data frames. Note that these negotiated keys are dependent on the MAC addresses of the client and victim, but this does not impact the attack, because the MC-MitM allows the adversary to create a rogue clone of the AP using the same MAC address as that of `WrongAP`.

In stage ③ of the attack, the client is connected to `WrongAP`. The attacker has to forward all traffic between the client and AP, and rewrite `WrongNet` to `TrustedNet` so the client keeps thinking `TrustedNet` is indeed nearby.

Whether the authentication succeeds depends on the protocol, more precisely on whether the SSID is used to derive the pairwise master key or session keys. If this is the case, then the attacker can no longer passively forward the 4-way handshake messages between the client and `WrongNet`, because they would derive different keys. Table 1 shows which protocols are (not) vulnerable.

To conclude, when a vulnerable protocol is used, a victim that was intending to connect with `TrustedNet`, will now instead successfully authenticate with and connect to `WrongNet`.

3.4 Home Network Authentication

3.4.1 WEP. The old WEP protocol is vulnerable. No PMK or session keys need to be computed, as WEP relies on a pre-shared key to directly encrypt all traffic. As a result, the attack works if two networks use the same pre-shared key.

3.4.2 WPA1/2. With WPA1 and WPA2, the PMK is derived from the pre-shared password and the SSID. Consequently, if a client is tricked into connecting to a network with a different SSID, it would use a different PMK, and therefore the 4-way handshake would fail.

3.4.3 WPA3. There are two possible versions of WPA3. The first one, which relies on SAE-loop, computes a group element P using a pre-shared key and the MAC addresses of the client and access point. P is then used to compute the PMK. The SSID does not influence the PMK, which makes the attack possible, as long as two networks use the same password. The second version relies on SAE-const, where the SSID is used to derive the PMK, so the attack fails, just like it does for home WPA1/2.

3.5 Enterprise Network Authentication

Enterprise network authentication relies on the EAP protocol. The derivation of the PMK is specified by each EAP method, such as EAP-MS-CHAPv2 or EAP-PWD. It is independent from the SSID for every single EAP method, meaning that it is vulnerable, as long as similar credentials are used. The passwords and usernames have to be the same, otherwise the attack fails. For certificates, it is less strict. For some devices, it suffices for the certificate to have the same CA. Devices can also check the CommonName or the public key.

3.6 Mesh Network Authentication

In a mesh network, peers first authenticate each other and negotiate a PMK using either SAE or 802.11X (recall Section 2.3). As detailed in the previous sections, all current EAP methods in 802.1X do not verify the SSID, and SAE only verifies the SSID when the latest SAE-const variant is used. Once a PMK is negotiated, the AMPE protocol is executed, which never verifies the SSID [1, §14.5.7]. As a result, mesh networks are vulnerable unless SAE-const is used.

3.7 Other Authentication Methods

3.7.1 Fast BSS Transition. The Fast BSS Rransition (FT) protocol is designed for client stations to be able to connect to different access points of the same network without the added latency of performing a full EAP handshake each time. FT uses a slightly modified 4-way handshake to establish a more dynamic key hierarchy. In particular, when connecting to the network for the first time, a PMK-R0 and PMK-R1 are calculated [1, §12.7.1.6.3]. The idea is that the central controller of the network, which manages all APs in the network, will store PMK-R0, and that each AP is given a different PMK-R1 [2].

The central controller calculates PMK-R0 by hashing a secret key derived from EAP authentication or from a pre-shared key, together with the network SSID and identifiers for the client and controller. Each access point can then calculate a PMK-R1 based on PMK-R0 and identifiers for the client and access point in order to perform the 4-way handshake when a client is roaming.

As a result, a client will only successfully complete the 4-way handshake when it is using the correct SSID, meaning that FT is not vulnerable to our SSID confusion attack.

3.7.2 FILS. FILS is another protocol designed to make the connection process faster. FILS public key authentication lets the client and AP authenticate one another using public keys. A shared secret is generated using Diffie-Hellman, and this shared secret is hashed together with a nonce generated by the client and a nonce generated by the access point to create the PMK. So whether the attack will succeeds depends on whether the client will trust the network, based on its public key. The standard specifies that clients can trust the network if they trust either the certificate authority or the AP's public key [1, §12.11.2.1].

For FILS shared key authentication, the client and AP already share a secret which they use to authenticate one another. This shared secret comes either from an EAP handshake or from a cached PMK. If it comes from the EAP handshake, then the client is vulnerable, as we discussed in section 3.5. Otherwise, the client is vulnerable only if they were already connected to WrongNet.

Table 2: Experiments against clients. Yes means the attack works against the protocols listed as vulnerable in Table 1.

	Attack Type	
Operating System	Standard	Optimized
Windows 11 (Lenovo Ideapad 5)	Yes	Yes
iOS 17.2.1 (Iphone 12)	Yes	Yes
Android v10 (Samsung Galaxy S9)	Yes	Yes
macOS 14.2.1 (MacBook Air M2 2022)	Yes	Yes

4 ATTACK OPTIMIZATION AND EVALUATION

In this section, we first evaluate our standard attack and then propose an optimized version of it. Additionally, we estimate against how many enterprise users our attack is possible.

4.1 Evaluation of the Standard Attack

We tested four devices with different OSes, shown in Table 2, against our attack using the EAP PEAP-MS-CHAPv2 protocol. The devices were tested by carrying out a full MC-MitM attack using two Wi-Fi dongles. We set up an access point called WrongNet, then set up a MitM that performs the attack. The clients then see TrustedNet in their list of available networks, even though TrustedNet is not present. After connecting to TrustedNet, the devices show that they are connected to TrustedNet on their GUIs, even though they are actually connected to WrongNet. This worked against all tested clients. We conjecture that all Wi-Fi clients are vulnerable.

In order to test devices using a single Wi-Fi dongle, we created a modified Hostapd version that simulates the MC-MitM position. This tool supports all major authentication methods: WEP, WPA2/3, WPA3 with SAE-Loop and SAE-Const, EAP, etc.

4.2 Optimized Connection-Only Attack

Most clients no longer check the SSID in received beacons once they are connected to a network. This means that the attacker only needs to be present while the victim is connecting to the network, and can then move the client back to the original channel. To test the feasibility of this attack, we first performed a full MC-MitM attack, and changed the SSID once the client has connected. When doing this, the client stays connected, indicating it was not checking the SSID in the beacons.

We also created a modified version of hostapd to test the optimized attack more easily using only a single Wi-Fi dongle. It can be started up with TrustedSSID as its SSID. After the tested client has connected, the Hostapd access point can be made to modify the SSID to WrongSSID in the beacons, probe, and association responses. To test whether the client stays connected after changing the advertised SSID, we repeatedly make the client negotiate a new group temporal key with the access point. If this negotiation succeeds, then we can safely assume that the client is still connected. If a client does not disconnect after changing the SSID, this implies it does not check the SSID once connected, and hence is vulnerable to our optimized attack variant. Using this tool, we tested all clients listed in Table 2, and all were found to be vulnerable.

4.3 Enterprise Evaluation

Our attack also applies to enterprise networks that use the same authentication settings. For instance, a university's Wi-Fi network may use the same authentication as eduroam. To investigate this, we scraped eduroam profiles and looked for SSIDs that use the same RADIUS server, revealing 6 vulnerable organizations: radius.vse.cz, nac.temple.edu, radius.york.ac.uk, radius.kuleuven.be, val.ul.ie, and eduroam.technion.ac.il. Employees of these universities, that only use their university's Wi-Fi network, can be tricked into connecting the (possibly less secure) eduroam network of another organization.

At our university, employees use the same Enterprise authentication settings to access the university's Wi-Fi and to connect to public hotspots throughout the country. These hotspots are broadcasted by the home routers of ISP's customers. Our attack can trick an employee into connecting to these hotspots, while the employee thinks they are connected to their university's network. This is problematic because the hotspot's owner is an ordinary ISP customer who can then intercept all traffic of the employee. Based on the scraped eduroam profiles, we found that employees of luxfuturelab.lu share authentication settings with Luxembourg's citiwifi hotspots, and are vulnerable to a similar attack.

5 DEFENSES

In this section, we propose, implement, and evaluate backwards-compatible defenses, and we propose updates to the 802.11 standard.

5.1 Improved Beacon Protection

Starting with Wi-Fi 7, which roughly corresponds to the IEEE 802.11be amendment, all APs must support beacon protection [6]. Beacon protection authenticates all transmitted beacons using a symmetric key. This key is called the beacon integrity group key and clients are given this key when connecting to the network. As a result, when beacon protection is used, a *connected* client can detect when an adversary changes the SSID in beacons. This leads to a possible defense against our SSID confusion attack: beacon protection must be enabled so that a client can verify the SSID *after* connecting to the network.

5.1.1 Current Limitations. Simply enabling beacon protection is unfortunately insufficient to prevent our attack. We experimentally confirmed this with hostap v2.10 on Linux kernel 6.6.14-1-lts when using virtual mac80211_hwsim interfaces: when beacon protection is enabled, it is still possible to make the victim connect to and send traffic towards an unintended network. With the standard attack, the client will drop all received beacons once connected, because the adversary is modifying the SSID in the authenticated beacon. This will eventually cause the client to disconnect due to beacon loss. However, even if the victim eventually disconnects, this can take several seconds during which the victim is vulnerable. Moreover, once the victim completes the 4-way handshake, the adversary can forward the real beacons without modifying them, since all tested implementations do not verify the SSID in the beacon once connected, ensuring the victim stays connected (recall Section 4.2).

The problem is that the client does not verify the authenticity of beacons that were received before connecting. Verifying the authenticity of previously-received beacons, at least once the beacon integrity group key is known, was nevertheless advised in the paper introducing beacon protection [15]. However, this check was not incorporated into the 802.11 standard [1].

5.1.2 Proposed defense. To prevent the attack, the first option is to let the client store the beacon from which the network's SSID was taken, and to verify the authenticity of this reference beacon during the 4-way handshake. More precisely, after receiving the beacon integrity group key in message 3 of the 4-way handshake, the client can verify the authenticity of the previously-captured reference beacon. If this succeeds, the handshake can be completed, and otherwise, the 4-way handshake must be aborted.

One aspect to take into account is that the beacon key might change between the time of receiving the reference beacon and receiving the key in the 4-way handshake, causing the handshake to fail. To remedy this, one option is to also let the AP transmit the previous (older) key, but that would require changing the functionality of the AP. A second option is to wait for a new beacon frame, verify the authenticity of the new beacon, and compare the static beacon content to the previously-captured reference beacon.

One downside is that the client must wait for a beacon before completing the 4-way handshake. Since most networks send a beacon every 102.4ms, this might cause noticeable delays. Nevertheless, this defense has the advantage that only the client needs to be patched, and that no protocol or major network configuration changes are required.

A less ideal solution would be to immediately complete the 4-way handshake and verify the SSID of the first beacon, and disconnect if it does not match. Although this would prevent delays when connecting to a network, an adversary can then block the legitimate beacon from arriving, meaning the client effectively remains vulnerable until it disconnects due to beacon loss.

5.1.3 Proof-of-Concept. We created a proof-of-concept of our defense by extending wpa_supplicant. The network is detected by passive scanning so that a reference beacon can be stored before connecting. During the 4-way handshake, the reference beacon is verified and the client disconnects if this fails. We confirmed that this prevents the attack.

5.1.4 Limitations. One limitation is that beacon protection cannot prevent our attack against hidden networks. In a hidden network, beacons do not contain the SSID of the network, and therefore beacon protection cannot be used to securely learn the SSID. However, nowadays it is no longer advised to use a hidden Wi-Fi network, because doing so forces clients to periodically send probe requests containing the network's SSID, which lowers the user's privacy [5].

5.2 Protocol Updates

Arguably the most reliable defense is to update the 802.11 standard to always authenticate the SSID when connecting to a protected network. This can be achieved by updating the 4-way handshake to either: (1) always include the SSID in the key derivation similar to FT; or (2) include the SSID as additional authenticated data in the handshake so clients can securely and easily verify the SSID. The second option can be implemented in a backward-compatible manner by including the SSID as an Information Element in the handshake's additional authenticated data. Old clients would ignore this element while new clients can use it to securely verify the SSID.

5.3 Avoiding Credential Reuse

Current networks can prevent the attack by avoiding credential reuse between different SSIDs, for instance, ensuring that different Enterprise networks use a different CommonName for the RADIUS server. Similarly, home networks can use different passwords for different SSIDs. Unfortunately, this would decrease usability when a network uses different SSIDs for the 2.4 and 5 GHz band, since then each SSID would require a different password or CommonName.

6 RELATED WORK

The closest related work is a combination of the KARMA attack with WPA2 password brute-forcing. In the KARMA attack, the adversary monitors the victim's probe requests to learn the names of the networks it connects to [4]. The adversary can then create a WPA2 network with that name and capture the first two messages of the 4-way handshake. These two messages are sufficient to perform an offline brute-force attack on the WPA2 password [7]. In case a weak password is used the adversary can recover the password and subsequently impersonate the network. In contrast, in our attack, we do not need to cover the victim's credentials, and our attack also works against different handshakes, including Enterprise networks.

Cassola et al. attacked enterprise WPA2 networks by creating a rogue clone of the network, where the clone's SSID included extra invisible characters [3]. The user then had to manually (re-)connect to this network and accept the new network certificate. AirEye used format strings to create a different SSID that appears identical to the real SSID in user interfaces [11]. When a user then manually connects to a network they might unknowingly select the wrong SSID to connect with. Stute et al. abused a flaw in Apple's Wi-Fi Password Sharing protocol to make the victim use an attacker-provided password instead of the real password [12]. Antonioli et al. studied Nearby Connections on Android and found that an attacker can instruct a peer to switch to a (different) Wi-Fi network by giving the peer the SSID and password of the new network [9]. This enabled them to intercept the peer's Internet traffic.

Vanhoef and Robben presented a method to create two WPA3-PK hotspots with different SSIDs but the same password [18]. Although this might, in theory, enable an SSID confusion attack, this requires that the victim first manually connects to the attacker's network, and also relies on implementation-specific parsing vulnerabilities.

7 CONCLUSION

We showed that users, or their apps, make security-sensitive decisions based on the network they are connected to. For instance, some VPNs can disable themselves when connected to a trusted Wi-Fi network. However, we demonstrated that a client can be tricked into unknowingly connecting to a different network, even when enterprise or home WPA3 protection is used. This is caused by a design flaw in several authentication methods defined in the 802.11 standard. This vulnerability was assigned CVE-2023-52424.

A backwards-compatible defense is to use beacon protection and to verify the authenticity of a beacon, and the network name contained in it, before exchanging data frames. Alternatively, the 802.11 standard can be updated to always authenticate the network name when connecting to a network.

ACKNOWLEDGMENTS

This research is partially funded by the Research Fund KU Leuven, and by the Cybersecurity Research Programme Flanders.

REFERENCES

[1] IEEE Std 802.11. 2020. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spec.*

[2] Devin Akin and David Coleman. 2008. Robust Security Network (RSN): Fast BSS Transition (FT). Retrieved 1 Feb. '24 from cwnp.com/uploads/802-11_rsn_ft.pdf.

[3] Aldo Cassola, William K Robertson, Engin Kirda, and Guevara Noubir. 2013. A Practical, Targeted, and Stealthy Attack Against WPA Enterprise Authentication.. In *NDSS*.

[4] Dino A Dai Zovi and Shane A Macaulay. 2005. Attacking automatic wireless network selection. In *IEEE SMC Information Assurance Workshop*. IEEE.

[5] Frederik Goovaerts, Gunes Acar, Rafael Galvez, Frank Piessens, and Mathy Vanhoef. 2019. Improving privacy through fast passive wi-fi scanning. In *NordSec*.

[6] Po-Kai Huang. 2022. 11be D2.0 CR for Beacon Protection. Retrieved 5 Feb. '24 from https://mentor.ieee.org/802.11/documents.

[7] Robert Moskowitz. 2003. Weakness in passphrase choice in WPA interface. Retrieved 1 Feb. '24 from wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html.

[8] Sritam Paltasingh. 2017. *Commissioning In Ad-Hoc Wi-Fi MESH Networks*. Master's Thesis. Eindhoven University of technology.

[9] Kasper Rasmussen, D Antonioli, and N Tippenhauer. 2019. Nearby threats: Reversing, analyzing, and attacking Google's 'nearby connections' on Android. In *NDSS*. Internet Society.

[10] Domien Schepers, Aanjhan Ranganathan, and Mathy Vanhoef. 2021. Let numbers tell the tale: measuring security trends in Wi-Fi networks and best practices. In *ACM WiSec*.

[11] Amichai Shulman. 2021. The SSID Stripping Vulnerability: When You Don't See What You Get. Retrieved 24 March '24 from https://aireye.tech/2021/09/13/the-ssid-stripping-vulnerability-when-you-dont-see-what-you-get/.

[12] Milan Stute, Alexander Heinrich, Jannik Lorenz, and Matthias Hollick. 2021. Disrupting continuity of apple's wireless ecosystem security: New tracking, DoS, and MitM attacks on iOS and macOS through bluetooth low energy,AWDL, and Wi-Fi. In *USENIX Security*.

[13] Manesh Thankappan, Helena Rifà-Pous, and Carles Garrigues. 2022. Multi-channel man-in-the-middle attacks against protected wi-fi networks: A state of the art review. *Expert Systems with Applications* (2022), 118401.

[14] Mathy Vanhoef. 2021. Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation. (2021).

[15] Mathy Vanhoef, Prasant Adhikari, and Christina Pöpper. 2020. Protecting Wi-Fi Beacons from Outsider Forgeries. In *ACM WiSec*.

[16] Mathy Vanhoef, Nehru Bhandaru, Thomas Derham, Ido Ouzieli, and Frank Piessens. 2018. Operating channel validation: Preventing multi-channel man-in-the-middle attacks against protected Wi-Fi networks. In *ACM WiSec*.

[17] Mathy Vanhoef and Frank Piessens. 2017. Key reinstallation attacks: Forcing nonce reuse in WPA2. In *ACM CCS*.

[18] Mathy Vanhoef and Jeroen Robben. 2024. A Security Analysis of WPA3-PK: Implementation and Precomputation Attacks. In *ACNS '24*.