On the Robustness of Wi-Fi Deauthentication Countermeasures

Domien Schepers
Northeastern University
schepers.d@northeastern.edu

Aanjhan Ranganathan
Northeastern University
aanjhan@northeastern.edu

Mathy Vanhoef
imec-DistriNet, KU Leuven
Mathy.Vanhoef@kuleuven.be

ABSTRACT

With the introduction of WPA3 and Wi-Fi 6, an increased usage of Wi-Fi Management Frame Protection (MFP) is expected. Wi-Fi MFP, defined in IEEE 802.11w, protects robust management frames by providing data confidentiality, integrity, origin authenticity, and replay protection. One of its key goals is to prevent deauthentication attacks in which an adversary forcibly disconnects a client from the network. In this paper, we inspect the standard and its implementations for their robustness and protection against deauthentication attacks. In our standard analysis, we inspect the rules for processing robust management frames on their completeness, consistency, and security, leading to the discovery of unspecified cases, contradictory rules, and revealed insecure rules that lead to new denial-of-service vulnerabilities. We then inspect implementations and identify vulnerabilities in clients and access points running on the latest versions of the Linux kernel, *hostap*, IWD, Apple (i.e., macOS, iOS, iPadOS), Windows, and Android. Altogether, these vulnerabilities allow an adversary to disconnect any client from personal and enterprise networks despite the usage of MFP. Our work highlights that management frame protection is insufficient to prevent deauthentication attacks, and therefore more care is needed to mitigate attacks of this kind. In order to address the identified shortcomings, we worked with industry partners to propose updates to the IEEE 802.11 standard.

CCS CONCEPTS

• **Networks → Wireless access points, base stations and infrastructure**; *Mobile and wireless security*; *Denial-of-service attacks.*

KEYWORDS

IEEE 802.11, Wi-Fi, Deauthentication, Protected Management Frames

ACM Reference Format:
Domien Schepers, Aanjhan Ranganathan, and Mathy Vanhoef. 2022. On the Robustness of Wi-Fi Deauthentication Countermeasures. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '22), May 16–19, 2022, San Antonio, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3507657.3528548

1 INTRODUCTION

Wi-Fi deauthentication attacks target the communication between a client and a wireless access point with the aim of disconnecting the client from the network. The main goal of deauthentication attacks is to cause a denial-of-service and render the network useless to one or more clients. For example, an adversary can disconnect a surveillance device such as a Wi-Fi IP camera from its network to prevent any video feedback, or interfere in a drone-to-controller connection [10]. These attacks can be launched by anyone within range of the victim, and nowadays low-cost Wi-Fi "deauthers" are readily available for less than $10 [31]. Disconnecting a client from the network is an essential step towards successfully executing many attacks, in particular those targeting the connection process, with example attacks being [3, 6, 7, 16, 23, 24, 39]. For instance, offline dictionary attacks against the passphrase of a WPA2 network require the adversary to capture a client's 4-way handshake, and can be accelerated by capturing additional handshakes. To capture a handshake, the adversary can disconnect any client from the network such that it (automatically) reconnects and executes a new handshake. Key reinstallation attacks also target the handshake [36, 37], and deauthentication vulnerabilities make it easier to perform this attack. Finally, the kr00k vulnerabilities [27], related to key reinstallation attacks, allow an adversary to decrypt data and requires clients are disassociated from the network as well.

Given the increasing importance of mitigating deauthentication attacks, the IEEE has standardized new protocols and security mechanisms. Specifically, the IEEE 802.11w amendment standardized Wi-Fi Management Frame Protection (MFP) and enhances the security of robust management frames. The standard defined various protection mechanisms such as data confidentiality, integrity, origin authenticity, and replay protection. In part, the goal was to prevent an adversary from forcibly disconnecting a client from the wireless network. The Wi-Fi Alliance made MFP mandatory in the recently released WPA3 security specification [2, 12]. Furthermore, from recent surveys in late 2021, 4.84% of the encrypted networks now support MFP, an adoption rate that has been growing in recent years [29] and is expected to increase further.

In this paper, we investigate the robustness of the countermeasures against deauthentication attacks. We first analyze the standard, particularly the rules for processing deauthentication and disassociation frames, on completeness, consistency, and security. We discover several contradictory rules and undefined (edge) cases, and we identify insecure rules that lead to new denial-of-service vulnerabilities. For example, we uncover new attacks during the connection phase. Furthermore, we inspect client and access point implementations of widely-used wireless daemons and operating systems, and identify a wide-variety of deauthentication attacks against the latest Linux kernel and implementations of *hostap*, IWD, Apple (i.e., macOS, iOS, iPadOS), Windows, and Android. Our identified vulnerabilities exploit flaws in the 4-way handshake implementations, the processing of beacon frame information elements in the kernel, and IEEE 802.1X authentication for enterprise networks. For example, an adversary can spoof a beacon frame that advertises an invalid bandwidth configuration which triggers the victim's

WiSec '22, May 16–19, 2022, San Antonio, TX, USA

ACM ISBN 978-1-4503-9216-7/22/05…$15.00
https://doi.org/10.1145/3507657.3528548

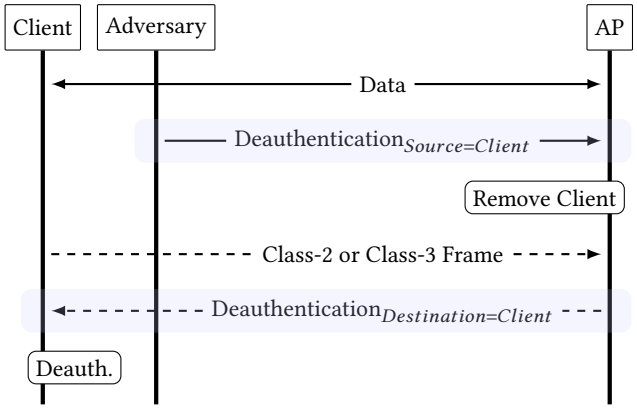

Figure 1: Classic deauthentication attack where an adversary spoofs a deauthentication frame from client to access point.

Linux kernel to disconnect from the network. Furthermore, we find implementation vulnerabilities that allow an adversary to replay protected management frames. Altogether, these vulnerabilities allow an adversary to disconnect any client from the network despite using MFP and are successful against personal and enterprise network configurations.

Our work highlights that management frame protection is insufficient to prevent deauthentication attacks, and therefore more care is needed to mitigate attacks of this kind. For example, our attacks exploiting unprotected beacon frames highlight the need for beacon frame protection as presented in [32]. Based on our findings, we collaborated with industry partners to propose updates to the standard, which have been presented at IEEE 802.11 meetings and will hopefully be adopted in future releases of the standard. Finally, we are making responsible disclosures to all affected vendors.

2 BACKGROUND

In this section, we present the classic deauthentication attack and the defenses offered by Wi-Fi Management Frame Protection (MFP).

2.1 Wi-Fi Deauthentication

Deauthentication attacks have existed since the early days of Wi-Fi. In a deauthentication attack, an adversary aims to disconnect a client from the network with the goal of disrupting the service or causing a persisting denial-of-service attack, rendering the network useless to any or all of its clients. Most commonly, the adversary leverages standardized deauthentication frames. A client typically sends these frames to indicate it is leaving the network. An access point can also transmit these frames when it is shutting down, or an error occurs. Unfortunately, an adversary can spoof the client or infrastructure and trivially inject deauthentication frames into the network, succesfully disconnecting the client from the network.

In Figure 1, we present the classic deauthentication attack in which an adversary spoofs a deauthentication frame from the client to the access point. In this attack, the client is not aware of its deauthentication until it sends a Class-2 (e.g., association management) or Class-3 (e.g., data) frame and the access points respond with a deauthentication frame (i.e., since the client was disconnected, its frames are no longer accepted). Similarly, the adversary can spoof a deauthentication frame from the access point towards the client station, resulting in its immediate disconnection from the network.

2.2 Management Frame Protection

Management Frame Protection (MFP), also called Wi-Fi Protected Management Frames (PMF), is defined in the IEEE 802.11w amendment and incorporated in the 2012 revision of the IEEE 802.11 base standard [17]. The standard provides protection mechanisms for management frames, including data confidentiality, integrity, origin authenticity, and replay protection. These protection mechanisms aim to increase the management frames' security and provide defense mechanisms against various known attacks that abuse management frames. For example, due to the protected deauthentication and disassociation frames, they prevent deauthentication attacks. Stations that have implemented the standard can indicate whether they are capable of using MFP, or whether they strictly require its usage. When a client station requires MFP, it will only establish a connection with APs that advertise its support. Similarly, APs will reject non-capable clients when their usage is mandatory.

2.2.1 Requirement in Certifications and Support. In recent years, the Wi-Fi Alliance required newly certified devices to support MFP and made its usage mandatory in modern network configurations such as WPA3 [2, 12]. To date, vendors such as Apple, Microsoft, Google, and the Linux kernel and various drivers such as Qualcomm (Atheros), Broadcom, and Intel support MFP. However, not all vendors and operating systems have yet implemented support; for example, to date, there is no support for MFP in FreeBSD and Linux drivers such as Realtek. From recent surveys in October and December 2021, 4.84% of encrypted networks supported MFP [29]. Interestingly, merely 0.01% of encrypted networks made its usage mandatory (i.e., MFP Capable and Required), a notable observation since its usage became mandatory in modern specifications such as WPA3. While its mandatory usage remains low, for example, due to the desire for backward compatibility, its general adoption rate is steadily growing over time. It is expected to grow even further with the adoption of WPA3 and Wi-Fi 6 [29].

2.2.2 Protection of Robust Management Frames. The robust management frames represent a subset of management frames that MFP can protect. The frames include robust action frames (e.g., spectrum management actions) and (re)association and (de)authentication frames. Notably, this excludes management frames such as beacons. The standard specifies operations for protecting both unicast and multicast (i.e., group-addressed) management frames. The standard protects against eavesdropping and forging for unicast management frames using the Counter Mode CBC-MAC Protocol (CCMP). Specifically, unicast frames are encrypted and authenticated using the Pairwise Transient Key (PTK) once this key is negotiated and installed for use. The standard also protects against forging using the Broadcast/Multicast Integrity Protocol (BIP) for multicast management frames. Specifically, multicast frames are authenticated but not encrypted using the Integrity Group Temporal Key (IGTK) once this key is installed. To support the protection mechanisms for unicast frames, the Robust Security Network Association (RSNA)

PTK establishment needs to be completed. Similarly, the IGTK has to be delivered to support the protection of multicast frames.

2.2.3 Security Association. The Security Association (SA) is a set of policies and keys used to protect the information in a connection and stored by each association party. When MFP is enabled, the security association is used to protect against forged (re)association frames, i.e., when a station has a valid security association, then (re)association requests will be temporarily rejected by the access point. Instead, the access point will initiate a SA Query procedure to determine the validity of the original SA. The procedure consists of a protected query and response between the stations, if completed successfully, verifies the SA is still valid, and the (re)association requests can be safely discarded.

3 ANALYSIS OF IEEE 802.11 MFP RULES

In this section, we study the rules in the IEEE 802.11 standard that specify how to handle deauthentication and disassociation frames in the context of MFP. This reveals undefined scenarios, uncovers contradictory rules, and reveals flaws in the standard that lead to Denial-of-Service (DoS) vulnerabilities. We practically verify all discovered vulnerabilities against the *hostap* daemon (Version 2.10).

3.1 Methodology and Overview

The latest version of the IEEE 802.11 standard centralizes most rules on how to handle (unexpected) deauthentication and disassociation frames in a single section [18, §12.6.19]. For ease of reference, these rules are listed and numbered in Appendix A, so that we can refer to a specific rule in the standard using the notation *rule* (*n*).

3.1.1 Threat model. We analyze the security of these rules in the context of an adversary within radio range of a victim. The victim can either be a client or an access point. We assume the adversary does not possess the credentials of the network. The adversary can spoof plaintext frames and can selectively block selected frames. Furthermore, the adversary can send a protected deauthentication or disassociation frame with an invalid authentication tag (Section 2.2.2). Sending such frames with a valid authentication tag is impossible because the adversary does not possess the correct keys.

3.1.2 Methodology. To study the completeness of the rules, i.e., whether all cases are defined, and to analyze their security, we created a table to summarize all rules in the standard (see Table 1). We focus on the handling of deauthentication and dissociation frames, since improperly handling those leads to DoS vulnerabilities. Because the rules in the standard do not treat clients and APs separately, we also do not explicitly differentiate their behavior. With this in mind, the last four columns in Table 1 contain the behavior of the station as specified in the standard. This behavior depends on the frame's destination address (first column), whether the corresponding PTK or IGTK has been installed (second column), whether the frame is protected (third column), and is dependent on the MFP support of the station and its peer (last four columns). Note that a station will possess the PTK once connected to a protected network, even if the station or peer does not support MFP. In contrast, a device will only possess the IGTK if it supports MFP. In the last four columns, *station* refers to the device whose receive and transmit behavior is described in the table, and *peer* refers to the client or AP that is connected to this station. The station's behavior depends on the MFP support of both the station and the peer, leading to the following four cases:

- Required: the station requires MFP support. This implies that the link between the station and peer always uses MFP.
- Both capable: the station supports MFP but does not require it. The peer supports or requires MFP, meaning MFP is used.
- Peer not capable: the station supports MFP but does not require it. The peer does not support MFP, meaning MFP is not used between the station and peer.
- Station not capable: the station does not support MFP. This means MFP is not used even though the peer might support (but not require) MFP.

Combined, this results in a complex landscape where many factors influence how a station handles deauthentication and disassociation frames. We also remark that the transmit behavior is not always explicitly defined in the standard but in some cases can be derived from the reception rules. In particular, when the standard specifies that a station should discard a frame, it also implies the peer should not transmit such frames. The resulting overview in Table 1 highlights that several cases are left undefined by the standard (marked with a dagger † symbol), meaning it is unclear how a station should behave in those cases, and that the behavior in some cases leads to DoS vulnerabilities. In the next sections, we will discuss these cases in detail. Where applicable, we also confirm attacks in practice. This is done against Linux's open source *hostap* daemon (Version 2.10).

3.2 Handling Unicast Frames

In this section, we study how stations handle unicast frames; that is, we discuss the rules as presented in the first four rows of Table 1.

3.2.1 Unprotected frames during handshake. The first row in Table 1 describes the behavior of stations when an unprotected unicast deauthentication or disassociation is received before the PTK has been installed. In the current standard, both rule (3) and (4) state:

> "The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK and IGTK are installed."

This leads to a denial-of-service attack while connecting to the network, even when MFP is in use. An adversary can trivially inject deauthentication or disassociation frames to abort the handshake. To prevent such an attack, a station should not immediately disconnect when a deauthentication or disassociation frame is received while connecting. Instead, we recommend that the receiver starts a timer on the reception of such a frame. The station can disconnect when there is no handshake progress before the timer expires. If the handshake does progress, i.e., the next frame is received, this timer is stopped, meaning the unprotected deauthentication or disassociation frame is effectively ignored.

We verified the attack against the latest *hostap* daemon and implemented our defense on top of it. Note we publish our proof-of-concept attack and defense as described in Section 5.1.4. Specifically, we extended the access point to start a timer when a deauthentication frame is received. When the timer expires, the deauthentication frame is processed as usual. On reception of an EAPoL frame, the timer is stopped, that is, the deauthentication frame is effectively

Table 1: Handling of deauthentication and disassocation frames using MFP, according to the IEEE 802.11 standard (Section 3.1). The first three columns indicate the receiver address, whether the station has the PTK or IGTK, and whether the frame is protected. The next four columns specify the behavior of the station depending on its MFP support, i.e., whether the station will accept (rx), transmit (tx), discard the frame, or initiate a SA Query procedure. For each behaviour the matching rule in the standard is included between parenthesis (see Appendix A).

			MFP Support			
Receiver	Key Available	Protected	Required	Both Capable	Peer Not Capable	Station Not Capable
Unicast	No	No	rx (4)	rx (3)	tx/rx (2)	tx/rx (1)
		Yes	discard †	rx (3)	discard (2)	discard (1)
	Yes (PTK)	No	SA Query †	discard (3)	tx/rx (2)	tx/rx (1)
		Yes	tx/rx †	tx/rx (3)	discard (2)	discard (1)
Group	No	No	discard (5)(6)	discard (5)	rx †	tx/rx (7)
		Yes	discard (5)(6)	discard (5)	discard †	rx (7)
	Yes (IGTK)	No	discard (5)(6)	discard (5)	discard †	—
		Yes	tx/rx †	tx/rx †	tx †	—

Legend: Vulnerable cases are highlighted in red. Undefined cases are marked with a dagger (†) and given their secure behavior.

ignored. We confirmed that spoofing deauthentication frames while the victim is connecting no longer results in a DoS. Our defense works when MFP is used and even when the network does not support MFP. Note that a similar defense can also be implemented by the client.

3.2.2 Protected frames during handshake. The second row in Table 1 describes how a station should react when a unicast protected frame is received during the handshake, i.e. when no PTK has yet been installed. This can occur when the client has already sent the last 4-way handshake message and installed the PTK, but the AP has not yet received this message. Additionally, as mentioned in our threat model, an adversary can send a protected frame with an invalid authentication tag. A first observation is that in this scenario, the standard does not specify how to handle protected deauthentication or disassociation frames when the network requires MFP (see the second row, column four). The secure behavior is to discard them. When both stations are MFP-capable, but MFP is not required, rule (3) applies, which in simplified terms states:

> "A STA [that is MFP-capable but does not require MFP] shall transmit and receive protected individually addressed robust Management frames to and from any associated STA that [is MFP-capable]."

Recall that deauthentication and association frames are robust management frames. This rule is not conditional on installing the PTK, making it unclear how this frame should be handled when no PTK is installed. Analogous to rule (7) we can assume that an implementation might then ignore the protection and not verify the authentication tag but still process the frame. This leads to a DoS vulnerability while the client is connecting, where the adversary can inject deauthentication or disassociation frames with an invalid authentication tag.

3.2.3 Incomplete handshake DoS. Inspired by the previous two DoS vulnerabilities during the connection process, we further analyzed the handshake and discovered a new DoS attack against the 4-way handshake. This attack is illustrated in Figure 2, where the adversary blocks the initial message 4 from arriving at the AP. As a result, the AP will eventually retransmit message 3. However, the handshake is completed from the client's perspective, so the client will install the PTK. This means the client will drop unprotected data frames, ignoring the retransmitted message 3 because it is transported in a plaintext data frame. Eventually, this will cause the handshake to timeout, resulting in a denial-of-service attack. We confirmed this attack in practice against the latest *hostap* daemon. This vulnerability is especially problematic in modern networks, where clients more often roam between various APs and hence more often need to execute the 4-way handshake.

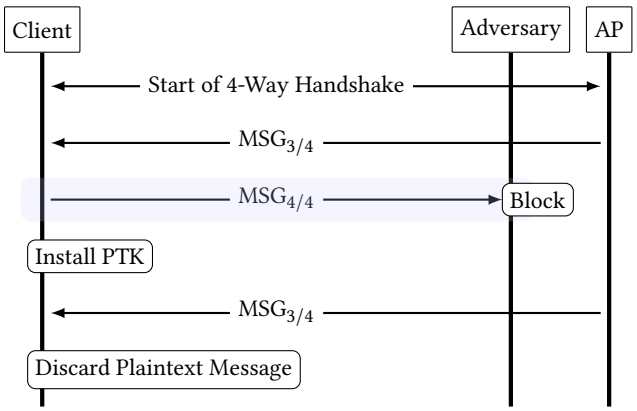

Figure 2: Blocking the 4th message in the 4-way handshake causes a retransmission of the 3rd message. Since the frame is in plaintext it is discarded, resulting in a denial-of-service.

Further complicating the situation is that the 4-way handshake may also be performed to refresh the PTK. When that is done, the 4-way handshake messages are protected using the current PTK. This implies that when message 4 is lost, the AP will retransmit

message 3 under the old PTK while the client is using the new PTK. Again, this results in a DoS vulnerability since the client will reject the retransmitted message 3. Although this attack during a rekey can be prevented by relying on the extended key ID for individually addressed frames features [18, §12.6.21], few devices support this, and the DoS attack during the initial 4-way handshake would remain possible.

To prevent a blocked or dropped message 4 from causing the initial 4-way handshake to timeout, we recommend that clients accept plaintext handshake frames until the first protected data frame is received. The idea is that once the peer transmitted a protected data frame, it must have completed the handshake, making it safe to no longer accept plaintext handshake frames.

3.2.4 Unprotected frames after handshake. The third row in Table 1 highlights two issues on how to handle the reception of unicast unprotected deauthentication or disassociation frames when the PTK has been installed. First, when MFP is required, the behavior of a station is undefined. The receiver should initiate a SA Query procedure to verify whether the alleged sender is still responsive to prevent attacks. When MFP is not required, but supported by both stations, rule (3) specifics that the deauthentication of the disassociation frame should be discarded. This contradicts other parts of the standard, which state that in this scenario, a SA Query procedure must be initiated to see whether the peer is still responsive [18, §11.13]. This is necessary to securely detect when the peer lost its keys, for instance, due to reboot, so the peer can smoothly reconnect to negotiate new keys.

3.2.5 Legitimate protected frames. The fourth row of Table 1 highlights an undefined case when MFP is required (row four, column four). The secure behavior can be derived from context: when both stations have a PTK, they must send and receive protected unicast deauthentication and disassociation frames.

3.3 Handling Group Frames

The last four rows in Table 1 specify how to handle group-addressed deauthentication and disassociation frames according to the standard. Notice that these frames are discarded, when MFP is required or both stations are capable, and when the IGTK is not yet installed or when the frames are not protected. This situation occurs when connecting to a network and when waking up from Wireless Network Management (WNM) sleep mode, where no IGTK will be installed in both cases. Additionally, a station that is not MFP capable will never possess an IGTK, resulting in two impossible combinations represented using a dash (see the last two rows).

3.3.1 Peer not capable undefined behavior. From Table 1 we can see that the handling of group-addressed deauthentication or disassociation frames is undefined when the station is MFP capable, but the peer is not. This case is further complicated because the behavior of a station will depend on whether it is acting as a client or AP. To derive the correct and secure behavior in this situation, we focus on infrastructure networks, where we can assume that only an AP will transmit frames with a group receiver address, and clients only process frames with a group receiver address. This allows us to determine when and how group-addressed frames should be transmitted or received:

- When a client is MFP capable but the AP is not, then the client will accept unprotected group-addressed deauthentication or disassociation frames.
- When an AP is MFP capable, it will have generated an IGTK key, and it should transmit group-addressed management frames using protection, even if some of its clients are not MFP capable. According to the standard this does not introduce issues, because rule (7) specifies that a station that does not support MFP, or did not enable it, should ignore the protection on received group-addressed robust management frames and hence treat it as an unauthenticated frame.

In all other cases, group-addressed deauthentication or disassociation frames should be discarded, leading to the listed secure behavior in Table 1 when the peer is not MFP capable. To avoid possible implementation vulnerabilities, the standard should be updated to define the above behaviors explicitly (see Section 5.2).

3.3.2 Protected group-addressed frames when in possession of an IGTK. The last row in Table 1 highlights the secure behavior when an IGTK has been installed, which implies that the station and peer support MFP. This behavior is not defined in the standard. In ordinary networks, the secure behavior is to transmit and accept protected group-addressed deauthentication and disassociation frames. Because the IGTK is shared between all network users; opening the possibility of insider attacks (as we will discuss in Section 5.1.3). To prevent insider attacks, the AP can choose to distribute a random IGTK to every client, or clients can ignore protected group-addressed deauthentication or disassociation frames.

4 WI-FI MFP IMPLEMENTATION ANALYSIS

In this section, we inspect wireless daemons and operating systems for their implementation of MFP and systematically search for novel deauthentication attacks bypassing the provided countermeasures.

4.1 Methodology and Experimental Setup

We first discuss the assumptions of the adversary within our threat model, followed by our research methodology. Next, we present the experimental setup used for our evaluation.

4.1.1 Threat Model. We consider an adversary with the goal of disconnecting any or all client stations from a network in which the usage of management frame protection is negotiated and enforced. To achieve this, the adversary can target both the client station or access point (i.e., by transmitting frames towards either station, potentially impersonating the other). We assume the adversary has no prior knowledge about the network; the adversary does not know any passphrases or cryptographic keys. Notably, this assumption implies we do not consider any insider threats for our implementation analysis; we discuss this threat in Section 5.1.3. Furthermore, the adversary can arbitrarily transmit, eavesdrop, intercept, record, and replay radio signals, as is commonly assumed to assess the security of wireless protocols. The adversary targets its victims solely on the Medium Access Control (MAC) layer and does not leverage any physical-layer techniques (e.g., jamming the channel) to disconnect the client. In this paper, we demonstrate all attacks can be executed using commercial off-the-shelf hardware (e.g., a low-cost Wi-Fi adapter supporting monitor mode and frame injection). We

assume that the adversary executes the attacks without physically tampering with any legitimate stations or modifying their firmware or driver code. Finally, we place no restrictions on the physical location of the adversary; the adversary can be anywhere within range of the radio signals.

4.1.2 Methodology. To investigate the robustness of the protections against deauthentication attacks, we start by examining open-source software to identify all references to deauthentication and disassociation calls. From these calls, we infer which module they occur and what functionality they perform. For example, we identify several deauthentication calls in Extensible Authentication Protocol (EAP) modules, responsible for handling the IEEE 802.1X authentication functionalities. Having identified potentially vulnerable modules, we search for a code execution path that triggers the deauthentication call. Particularly, we inspect frames that can be sent in plaintext to its receiver. We also inspect encrypted or integrity-protected frames that can be replayed, such that the adversary can spoof them without any network credentials (e.g., passphrases or encryption keys). Surprisingly, we find that many deauthentication calls can be triggered in ways that bypass all of the MFP protection mechanisms. Having understood structural and common flaws in open-source software, we aim to replicate them or identify variants on closed-source (operating) systems. Furthermore, our research focuses particularly on modern and upcoming network configurations such as WPA3 since these configurations made it mandatory for their clients to enforce MFP usage.

4.1.3 Experimental Setup. In our evaluation, we inspect daemons and operating systems that support MFP. In all experiments, we configure the network to enforce MFP usage. For *hostap*, we evaluate the widely-deployed Version 2.9 and latest Version 2.10 released in January 2022. For IWD, an upcoming daemon for Linux, we test Version 1.26. For the Linux kernel, we evaluated release Version 5.11.0-38-generic, as well as latest Long-Term Support (LTS) kernel Version 5.15.0-051500-generic released in November 2021. For Android, we use a Google Pixel 4 XL running Android 12, and a Xiaomi Mi 10T 5G and Samsung Galaxy Note 10 running Android 11. For Apple, we use a 2018 MacBook Pro on latest macOS Version 12.3, iPhone Xs Max on latest iOS Version 15.4, and a third generation iPad Pro on latest iPadOS Version 15.4. For Windows, we use an HP ZBook Power running Windows 10 build 19043.1466, using its default Intel AX201 network card and a TL-WN722N dongle. We use a TP-Link AC600 Archer T2UH, a low-cost commercial dongle supporting monitor mode and frame injection for the adversary.

Summary of Results and Replicability. In Table 2, we present a summary of all evaluated (operating) systems with respect to the deauthentication techniques identified in the remainder of this section. These techniques abuse frames of the 4-way handshake (Section 4.2), forge beacon frames (Section 4.3), and forge Extensible Authentication Protocol (EAP) and EAP over LAN (EAPoL) frames of IEEE 802.1X (Section 4.4). In Table 2, a dash indicates the respective technique does not apply to the evaluated system. Furthermore, for each of the deauthentication attacks, we write a test case (i.e., proof-of-concept) within the Wi-Fi Framework [30]. The attacks are straightforward to execute against any new system (e.g., Linux, Windows, Apple, Android). We publish all test cases (Section 5.1.4), enabling anyone to replicate our results and test for the weaknesses within their systems and devices.

Table 2: Summary of deauthentication techniques against systems enforcing Wi-Fi Management Frame Protection (MFP).

Technique	hostap-2.9	hostap-2.10	IWD 1.26	Linux 5.15.0	macOS 12.3	iOS 15.4	iPadOS 15.4	Android 12	Windows 10
Malformed Msg1	●	●	●	–	○	○	○	○	○
IGTK Installation	○	○	○	–	○	○	○	●	◐
Bandwidth Change	–	–	–	●	○	○	○	○	◐
Channel Switch	–	–	–	●	●	●	●	○	◐
EAPoL Logoff	●	●	–	–	○	○	○	○	–
EAP Failure	●	●	●	–	○	○	○	○	○
Max EAP Rounds	●	●	○	–	○	○	○	○	○
Max Re-Auths.	●	●	–	–	○	○	○	○	–

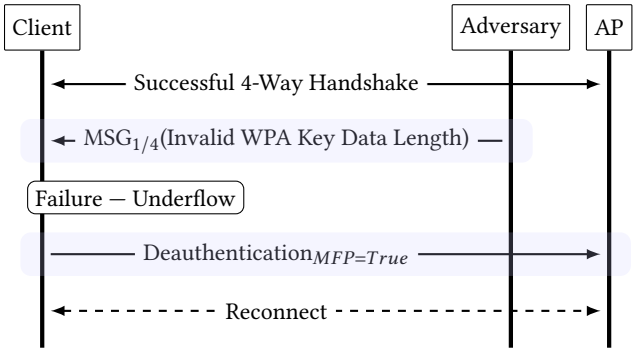

Figure 3: Deauthentication attack using an invalid WPA Key Data Length in 4-Way Handshake 1/4, causing an underflow.

4.2 4-Way Handshake

We first investigate modules responsible for the 4-way handshake. Specifically, we start by investigating potential deauthentication vulnerabilities that an adversary can exploit after successfully completing the 4-way handshake. This implies that these vulnerabilities can be exploited even if the adversary was not present while the victim was connecting to the network. We also investigate the handshake itself and test whether the Broadcast/Multicast Integrity Protocol (BIP) is correctly configured during the 4-way handshake.

4.2.1 Malformed 4-Way Handshake Message. We find clients can be forcibly disconnected by injecting a specially crafted Message 1 of the 4-way handshake. Upon receiving this message, the client will process all its fields. Our first attack variant includes a malformed key data field in Message 1. Under normal conditions, this key data field may be used to transport a Pairwise Master Key (PMK) ID, which the AP can include in Message 1 when PMK caching is configured (e.g., to support key caching while roaming between APs). However, since the first message of the 4-way handshake is

not protected, i.e., it is not encrypted nor authenticated, an adversary can include malformed key data. In particular, by setting the key data length field to a value that does not match the length of the transmitted data, we can trigger an underflow when the client processes the frame. In Figure 3, we present an overview of the attack using an invalid WPA Key data length in Message 1 of the 4-way handshake. Transmitting the frame towards the client with a malformed length value will cause the client to report an underflow when processing the element, forcing the client to send a protected unicast deauthentication frame towards the AP. The adversary can transmit this frame at any time after the victim client has established a successful connection with the network, as long as the victim accepts plaintext handshake message after it is connected. Recall from our standard analysis in Section 3.2.3 that (temporarily) accepting plaintext handshake messages after connecting is required to, under normal conditions, reliably handle retransmitted messages. We remark the network configuration is not required to use the key data field; an adversary can spoof the message with a bogus key data element having a malformed length value. We confirmed this attack against the supplicant of *hostap* Version 2.9 and 2.10 (Table 2), and is effective against any network configuration.

In our second attack variant, the adversary transmits a Message 1 with invalid key information flags. These flags are normally used to specify characteristics of the frame, e.g., whether it is encrypted or authenticated. In particular, by sending a Message 1 that incorrectly has the install flag set, the receiver will abort the handshake and disconnect. We confirmed this attack against the IWD daemon Version 1.26 (Table 2), and is effective against any network configuration.

4.2.2 Installation of Group Temporal Key Packet Number. When a client joins the network, it receives an Integrity Group Temporal Key (IGTK) Packet Number (IPN) during the 4-way handshake. The IPN is used within the Broadcast/Multicast Integrity Protocol (BIP) and prevents replay attacks of group-addresses (i.e., broadcast) frames (Section 2.2). As the standard requires, the client has to install the respective IPN. However, we find this does not always happen in practice. If the client fails to install the appropriate IPN, group-addressed robust action frames, such as deauthentication and disassociation frames, become subject to replay attacks. As a practical example, consider an AP which transmits a broadcast deauthentication frame using some IPN value. Then, a vulnerable client connects to the network, and since it does not install the appropriate IPN, it will instead set the packet number to zero. Now, an adversary can replay the old deauthentication frame since the frame's IPN exceeds the IPN configured by the vulnerable client. We confirmed this attack against clients of Android 11 and 12 (Table 2), and is effective against any network configuration. The attack also worked against Windows 10 when using the TL-WN722N dongle but not the Intel AX201 network card.

4.3 Beacon Information Elements

In Linux, *softmac* drivers manage the MAC Sublayer Management Entity (MLME) in software, in contrast to offloading this to the network card. For client stations, the MLME is implemented directly in the kernel (i.e., in the *mac80211* subsystem), and for access points, it is implemented in userspace (e.g., *hostap*). The MLME performs

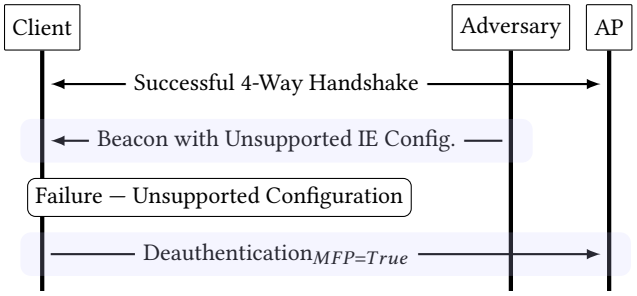

Figure 4: Deauthentication attack using a beacon advertising unsupported configurations in its Information Elements (IEs), causing the client's kernel to force a deauthentication.

various tasks such as connection management and handling management frames, including processing or transmission of beacon frames. However, while a beacon frame is a management frame, it is not a robust frame protected under MFP (Section 2.2), and thus remains at risk of manipulations by an adversary. To address this shortcoming, researchers proposed a method to protect beacon frames using a protected Management MIC Element [32]. To date, the defense mechanism has been implemented in the Linux kernel and *hostap*. However, to enable beacon protection, the hardware (and/or firmware) of a client's network card needs to be updated to support it. Recent surveys found none of the APs supported protected beacon frames (i.e., none of the beacon frames included a Management MIC Element) [29]. Until its adoption grows, which is expected to be slow given the adoption of new Wi-Fi standards [29], clients remain at risk to beacon manipulations and the deauthentication attacks presented in the remainder of this section.

4.3.1 Modifying the Physical-Layer Configuration. We find that setting invalid or unsupported parameters in the information elements of a beacon frame can force the MLME of a client to cause a disconnection. That is, when the client's kernel (MLME) processes physical-layer network information in beacons, it may fail to support a newly advertised configuration (e.g., switching to an alternative unsupported channel). An adversary can abuse this by spoofing beacon frames and making the necessary adjustments to cause a client to disconnect from the network even when management frame protection is enforced. Since this flaw exists in kernel space, it is independent of any userspace applications (e.g., *hostap* or IWD) and network configurations (e.g., personal, enterprise). In Figure 4, we present a generalized outline of how the adversary can cause the client to deauthenticate. Here, the adversary can simply replay a beacon frame and modify or add information elements, advertising a new and potentially invalid physical-layer configuration. By inspecting Linux's source code, we identified two information elements which can force a client to disconnect: the High Throughput (HT) Information element in which the access point specifies its HT information (e.g., its bandwidth configuration), and the Channel Switch Announcement (CSA) element in which the access point requests its clients to switch to a new channel configuration.

4.3.2 Bandwidth Configuration Change. In Linux, the client's kernel (i.e., MLME) processes any bandwidth configuration changes

and verifies if these configurations are supported and valid (e.g., due to its hardware capabilities and regulatory restrictions). For example, IEEE 802.11n allows for double the bandwidth (40 MHz instead of a default 20 MHz), which results in slightly more than double the data rate. An access point advertises any such capabilities in its High Throughput (HT) Capabilities and Information IEs in beacons. However, there are some limits on which channels can be used for the widened bandwidth. For example, HT40- (which defines both 20 MHz and 40 MHz with secondary channel below the primary channel), can only be used on channels 5-13 on the 2.4 GHz band. We find that when changing the bandwidth of the access point in the beacon frame to a value that the Linux kernel does not support, the client will fail to switch to the new configuration and disconnect from the network. As an example, consider a 40 MHz channel on the 2.4 GHz frequency band using IEEE 802.11g and IEEE 802.11n. In such a configuration, the access point will advertise a primary and secondary channel to accommodate for the 40 MHz bandwidth. In the HT Information IE, the access point indicates whether the secondary channel is above or below the primary channel. When an adversary corrupts this parameter, for example, to flip the secondary channel to the opposite (e.g., below instead of above), the client's kernel will not follow the requested bandwidth change. In turn, this failure results in the client disconnecting from the network and transmitting a protected deauthentication frame. We confirmed this technique proves successful against the Linux kernel in the latest Version 5.15.0 (Table 2), and is effective against any user-space application and network configuration. The attack also worked against Windows 10 when it used the Intel AX201 as a network card but not when it was using the TL-WN722N dongle. We conjecture that other parameters manipulating the bandwidth, for example, in the (Very) High Throughput information elements, will cause similar disconnection results when the requested change is invalid or not supported by the client's kernel.

4.3.3 Channel Switch Announcement. Announcing the switch to an unsupported (or non-existing) channel will cause the Linux kernel to force a deauthentication. Specifically, an adversary can add a Channel Switch Announcement (CSA) Information Element (IE) to a captured beacon frame and replay it only once. The client's kernel will then attempt to switch to the announced channel; however, if this channel is unsupported (e.g., due to regulatory restrictions), or the channel does not exist, then the switch will fail and trigger the deauthentication. In a similar approach, we find that Apple clients (macOS, iOS, and iPadOS) accept a channel switch to an *existing* channel. An adversary can use this to have the client switch to a channel on which the access point does not operate. Since the client receives no responses, it will switch to the original channel and attempt to re-authenticate and re-associate with the network. However, since a valid security association exists, its reassociation request is not accepted by the access point (Section 2.2.2). Apple clients will transmit a protected disassociation frame upon receipt of the rejection, thereby disconnecting from the network. Note the reassociation response is not protected and thus can be spoofed by an adversary. Depending on the configuration (i.e., if "Auto-Join" is enabled on Apple clients), the client may attempt to reconnect to the network. We confirmed this technique proves successful against the Linux kernel in latest Version 5.15.0, and Apple (macOS 12.3, iOS 15.4, iPadOS 15.4) clients (Table 2), and is effective against any user-space application in Linux and network configuration. The attack also worked against Windows 10 when using the Intel AX201 network card but not when using the TL-WN722N dongle.

We conjecture that IBSS (Independent Basic Service Set) networks (i.e., ad-hoc or peer-to-peer networks that operate without access points) in Linux are vulnerable to this attack technique as well. However, while the standard provisions support for Wi-Fi MFP in IBSS networks, we, unfortunately, were unable to identify and evaluate a practical setup that implemented support for ad-hoc networks enforcing protected management frames. Executing the attack against the non-protected IBSS network resulted in the deauthentication and reconnection of the victim client.

4.3.4 Discussion. While channel-switch approaches have previously been described in the context of denial-of-service attacks [19], we found that they remain possible even when MFP is being used. Although it is also possible to transmit a CSA inside a unicast action frame, which MFP would protect, this does not scale when many clients are connected to the AP since a separate CSA would have to be sent to each client. As a result, most networks are expected to use beacons to advertise a channel switch in practice. Combined, our techniques bypass any management frame protection when leveraging the beacon frame (unprotected).

4.4 IEEE 802.1X Authentication

The third module we analyze for implementation vulnerabilities is responsible for IEEE 802.1X authentication. Specifically, we investigate how any of the Extensible Authentication Protocol (EAP) and EAP over LAN (EAPoL) frames can be leveraged to impact the authentication state machines to end up in a disconnected or failing state. When IEEE 802.1X is in use, we find there are numerous methods of terminating the authentication procedure or purposefully failing it, such that the client station gets disconnected from the network. Since the modules in this section are responsible for IEEE 802.1X, the attack techniques of this type target enterprise network configurations only. We confirmed all techniques in this section to be successful against *hostap* Version 2.9 and 2.10 (Table 2).

4.4.1 EAPoL Logoff. Upon receiving an EAPoL Logoff frame, the AP (i.e., authenticator) will restart an IEEE 802.1X authentication procedure. However, a scheduled disconnect *after EAP-Failure* will be configured, consisting of a timeout interval after which the AP will disconnect the client (e.g., due to failure to complete a new authentication session successfully). In *hostap*, we find this timeout interval is configured at 10 milliseconds. Practically, we find this timeout interval is always exceeded. In Figure 5, we present an overview of the attack where an adversary injects the EAPoL Logoff frame, causing the AP to timeout and transmit a protected deauthentication frame towards the targeted client. Notably, we find the client disables the SSID (i.e., network) for 10 seconds upon receiving the deauthentication frame; the client temporarily blocks the network and waits to reestablish a new connection.

4.4.2 EAP Failure. Similar to the EAPoL Logoff attack technique, an adversary can target the client by transmitting an EAP Failure frame. This frame informs the client that a failure has occurred within the authentication procedure, and the procedure will be

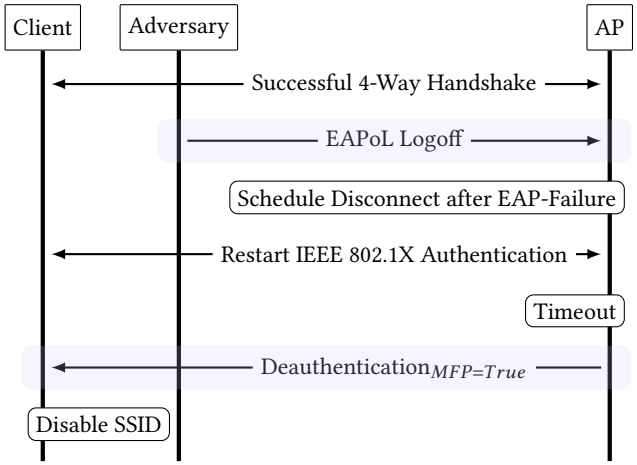

Figure 5: Deauthentication attack using an EAPoL Logoff frame, causing the access point to configure a timeout which disconnects the client before a re-authentication completes.

terminated. Since the client has already established a connection with the AP, it now expects a (protected) deauthentication frame. However, since an adversary spoofed the AP (i.e., by sending the EAP Failure), the AP will send no deauthentication frame. To prevent getting stuck, the client configures an authentication time-out interval, after which it will disconnect by sending a protected deauthentication frame towards the AP. In *hostap*, we find this time-out interval is configured at 2 seconds. When triggered, the client disables the SSID (i.e., network) for 10 seconds. Furthermore, the attack is successful against IWD, immediately disconnecting the client.

4.4.3 Maximum Number of EAP Authentication Rounds. While RFC 4137 does not put any limit on the number of messages in an authentication session, client implementations do configure an upper limit in practice to avoid potential loops with authentication servers. In *hostap*, we find the client limit (i.e., `EAP_MAX_AUTH_ROUNDS_SHORT`) is configured at 50 rounds. An adversary can now inject EAP frames towards the client (e.g., EAP Identity Requests), such that the upper limit gets exceeded. Upon exceeding the limit, the client aborts the authentication session, disconnects, and transmits a protected deauthentication frame to the AP. Similarly to the client, the AP configures an upper limit. However, in *hostap*, we find the AP implements a state model which does not increase its counter (set to an upper limit as defined by the `max_auth_rounds` parameter in its configuration file) after a successful connection has been established. As a result, *hostap*'s AP is not vulnerable to this attack.

4.4.4 Maximum Number of Re-Authentications. Similar to the maximum number of EAP authentication rounds, there is a maximum number of re-authentications that an AP accepts. In *hostap*, the limit (i.e., `reAuthMax`) is configured at two. An adversary can now spoof the client and inject three EAPoL Start frames such that the limit gets exceeded. Upon exceeding the limit, the AP schedules a disconnect *after EAP-Failure*, times out, disconnects, and transmits a protected deauthentication frame to the client. Upon receipt, the client disables the SSID (i.e., network) for 10 seconds.

5 DISCUSSION AND COUNTERMEASURES

In this section, we discuss our findings and present countermeasures to increase the overall robustness against deauthentication attacks. By collaborating with industry partners, we also propose updates to the IEEE 802.11 standard in order to address our discovered flaws.

5.1 Discussion

Our findings revealed that current mitigations are insufficient to prevent deauthentication attacks, expose vulnerabilities that lead to a denial-of-service, and insufficiently address insider threats.

5.1.1 Insufficient Mitigations to Address Deauthentication Attacks. Our analyses of the standard and its implementations have shown that MFP does not provide sufficient mitigations to protect against deauthentication attacks. For example, we identified the standard insufficiently addressed edge cases on how to handle the transmission and reception of robust management frames (e.g., whether an access point should protect robust management frame when it supports but does not require MFP). Furthermore, while the standard is designed to increase the security of management frames, we find data frames can be used in practice to perform deauthentication attacks (e.g., using IEEE 802.1X EAPoL frames as shown in Section 4.4). Therefore, protecting against a deauthentication attack goes beyond the protection of management frames, and thus more care is needed to protect against this type of attack.

5.1.2 Denial-of-Service. The identified attacks can target a specific victim client and any vulnerable client connected to the network. For example, the attacks in Section 4.2 and Section 4.4, which leverage the 4-way handshake and IEEE 802.11X respectively, will target a unique client since the adversary has to spoof frames from or towards said client, thereby forcing only the specific client to disconnect from the network. On the contrary, the attacks in Section 4.3, which leverage the beacon frame's information elements, may disconnect multiple clients from the network at once since it spoofs the access point and thereby will cause any vulnerable client station to disconnect from the network. In addition to disconnecting victim clients, our findings can be used for denial-of-service attacks. For example, the attacks based on IEEE 802.1X (Section 4.4) may cause a client to disable the network for a duration of 10 seconds. This can aid an adversary in achieving a longer-lasting denial-of-service and may be of use in an attack that requires that the victim client does not automatically and immediately reconnect to the network.

5.1.3 Insider Threats. In the threat model of our implementation evaluation (Section 4.1.1) we did not consider any insider threats. However, it is essential to note insider threats remain present, even when protected management frames are enforced. For example, an adversary who is connected to the network knows the integrity group key, which can be used to forge broadcast deauthentication or disassociation frames, forcing all the other clients to disconnect.

5.1.4 Repository and Responsible Disclosures. We publish our proof-of-concept attacks and countermeasures (Section 3) and test cases (Section 4) on GitHub [1]. Finally, we are disclosing the identified vulnerabilities to their respective vendors and track available patches.

[1] https://github.com/domienschepers/wifi-deauthentication

5.2 Countermeasures

In this section, we present countermeasures for the standard and its implementations to better defend against deauthentication attacks. Furthermore, we make network configurations recommendations.

5.2.1 Summary of Proposed Standard Updates. We believe that the root cause of several identified issues in the IEEE 802.11 standard is the complexity behind the rules. Although the rules can be written more clearly, some complexity inevitably remains as long as backward compatibility is needed. A long-term solution is, therefore, to mandate the usage of MFP for all devices in all protected Wi-Fi networks, which in turn would simplify how deauthentication and disassociation frames should be handled.

To address the discovered issues, we wrote updates to the 802.11 standard in collaboration with industry partners. Our changes are backwards-compatible, though we recommend to eventually make MFP mandatory as well. Concretely, we rewrote the rules to be more explicit, ensuring no undefined cases, and so they defend against all discovered attacks. The incomplete handshake attack of Section 3.2.3 is also highlighted in our proposed changes, with the concrete defense being optional and left for vendors to determine. Our proposed changes have been presented at TGm task group meetings of the IEEE 802.11 and were positively received [28]. As a result, we are hopeful that our suggestions, or slight variations thereof, will be included in the next version of the 802.11 standard.

5.2.2 Countermeasures for Implementations. Generally, we find it is more robust to silently discard any corrupted or invalid frames instead of disconnecting from the network. This approach would, for example, prevent an adversary from abusing malformed handshake messages (Section 4.2.1). Similarly, to defend against deauthentication attacks based on IEEE 802.1X (Section 4.4), it is essential to silently discard plaintext EAP and EAPoL frames when an MFP-protected connection has been established. This approach is taken by vendors such as Apple and Windows and hence are not vulnerable to this attack. However, care must be taken that this does not result in a DoS vulnerability when an adversary blocks message 4 from arriving, causing the plaintext retransmitted message 3 to be ignored and the handshake to fail (recall Section 3.2.3). Our recommendation is to drop plaintext EAP and EAPoL frames once a valid protected data frame has been received. This assures the handshake still completes, even if message 3 or 4 was lost, while still preventing an adversary from abusing plaintext EAP and EAPoL frames after the handshake is completed.

5.2.3 Recommendations for Network Configurations. Ideally, networks are configured to *require* protected management frames. Unfortunately, even though MFP was defined in 2009 and has now been adopted by most major (operating) systems, older non-MFP-capable devices would no longer be able to connect. Furthermore, the network should enable protected beacon frames [32], which to date has been implemented in the Linux kernel and *hostap* daemon. Beacon frame protection is backwards-compatible and will prevent outsider forgeries of beacons [32] as abused in Section 4.3.

6 RELATED WORK

In previous works, researchers identified denial-of-service attacks based on, for example, network handshakes in WPA2 [15, 35] and WPA3 [9, 38], the security capabilities of a network [21], channel switching mechanisms [19], and the authentication mechanisms of WPA3 [20] and IEEE 802.1X [11, 25]. In this paper, we identified novel denial-of-service vulnerabilities during the 4-way handshake, particularly when the usage of MFP is enforced. Furthermore, an adversary can perform a denial-of-service by jamming the wireless radio channel [4, 7, 26] and can be achieved even with low-cost commercial hardware [34]; physical-layer attacks were excluded from our threat model. In this paper, we investigated the robustness of countermeasures against denial-of-service and deauthentication attacks in the context of MFP. Prior research evaluated the MFP standard and identified denial-of-service attacks against the SA Query procedure in earlier drafts of the standard [1, 13] as well as its default timeout intervals in commercial systems [8], deadlock vulnerabilities [5, 14], and the 4-way handshake [40]. Furthermore, researchers evaluated the resilience against deauthentication and association flooding attacks [8], and implementation vulnerabilities in the *hostap* daemon allowed an adversary to trick the access point in deauthenticating all clients by transmitting association frames from invalid source addresses [30]. In this paper, we performed the first study of the standard, investigating how stations are expected to handle deauthentication and disassociation frames in the context of MFP. As a result, we identified novel denial-of-service and deauthentication attacks in both the standard and its implementations. For example, we presented attacks that forge information elements in the beacon frame (e.g., High Throughput Capabilities and Information elements). Our findings highlight the importance of detecting forged beacon frames [22] and adopting its latest protection mechanisms [32], as well as adopting the security mechanisms to prevent man-in-the-middle attacks [33].

7 CONCLUSION

In this paper, we inspected the standard and implementations of Wi-Fi Management Frame Protection (MFP), as defined in IEEE 802.11w-2009, for their robustness and protection against deauthentication attacks. In our analysis of the 802.11 MFP rules, we discovered unspecified edge cases, uncovered contradictory rules for processing robust management frames, and revealed insecure rules that lead to denial-of-service vulnerabilities. In our implementation analysis, we found the Linux kernel, *hostap* and IWD daemons, Apple (i.e., macOS, iOS, iPadOS), Windows, and Android are vulnerable to a variety of deauthentication attacks against personal and enterprise network configurations. Our findings enable an adversary to prevent a client from joining the network or disconnecting any client despite the enforced usage of management frame protection. As a result, we find usage of MFP is insufficient to protect against deauthentication attacks. We recommended countermeasures and worked with industry partners to propose updates to the IEEE 802.11 standard to address the identified shortcomings.

ACKNOWLEDGMENTS

This research is partially funded by the Research Fund KU Leuven, and by the Flemish Research Programme Cybersecurity. Mathy Vanhoef holds a Postdoctoral fellowship from the Research Foundation Flanders (FWO).

REFERENCES

[1] Md Sohail Ahmad and Shashank Tadakamadla. 2011. Short paper: security evaluation of IEEE 802.11 w specification. In *Proceedings of the fourth ACM conference on Wireless network security*. 53–58.

[2] Wi-Fi Alliance. 2021 (Accessed 17 November 2021). Security | Wi-Fi Alliance. https://www.wi-fi.org/discover-wi-fi/security.

[3] Alberto Bartoli, Eric Medvet, Andrea De Lorenzo, and Fabiano Tarlao. 2018. (In) Secure configuration practices of wpa2 enterprise supplicants. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*. 1–6.

[4] John Bellardo and Stefan Savage. 2003. 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions. In *USENIX security symposium*, Vol. 12. Washington DC, 2–2.

[5] Benjamin Bertka. 2012. 802.11 w security: DoS attacks and vulnerability controls. In *Proc. of Infocom*.

[6] Sebastian Brenza, Andre Pawlowski, and Christina Pöpper. 2015. 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*. 1–11.

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

[8] Efstratios Chatzoglou, Georgios Kambourakis, and Constantinos Kolias. 2021. Empirical Evaluation of Attacks Against IEEE 802.11 Enterprise Networks: The AWID3 Dataset. *IEEE Access* 9 (2021), 34188–34205.

[9] Efstratios Chatzoglou, Georgios Kambourakis, and Constantinos Kolias. 2022. How is your Wi-Fi connection today? DoS attacks on WPA3-SAE. *Journal of Information Security and Applications* 64 (2022), 103058.

[10] The MITRE Corporation. 2022 (Accessed 12 February 2022). CVE - CVE-2019-3944. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-3944.

[11] Ping Q Ding, JN Holliday, and Aslihan Celik. 2004. Improving the security of Wireless LANs by managing 802.1x Disassociation. In *First IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 53–58.

[12] Philipp Ebbecke. 2020 (Accessed 17 November 2021). Protected Management Frames enhance Wi-Fi network security | Wi-Fi Alliance. https://www.wi-fi.org/beacon/philipp-ebbecke/protected-management-frames-enhance-wi-fi-network-security.

[13] Martin Eian. 2009. Fragility of the robust security network: 802.11 denial of service. In *International Conference on Applied Cryptography and Network Security*. Springer, 400–416.

[14] Martin Eian and Stig F Mjølsnes. 2012. A formal analysis of IEEE 802.11 w deadlock vulnerabilities. In *2012 Proceedings IEEE INFOCOM*. IEEE, 918–926.

[15] Changhua He and John C Mitchell. 2004. Analysis of the 802.11 i 4-Way Handshake. In *Proceedings of the 3rd ACM Workshop on Wireless Security*. 43–50.

[16] 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. 2021. 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*. 1100–1117.

[17] IEEE Std 802.11. 2012. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*.

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

[19] Bastian Könings, Florian Schaub, Frank Kargl, and Stefan Dietzel. 2009. Channel switch and quiet attack: New DoS attacks exploiting the 802.11 standard. In *2009 IEEE 34th Conference on Local Computer Networks*. IEEE, 14–21.

[20] Karim Lounis and Mohammad Zulkernine. 2019. Bad-token: denial of service attacks on WPA3. In *Proceedings of the 12th International Conference on Security of Information and Networks*. 1–8.

[21] Karim Lounis and Mohammad Zulkernine. 2020. Exploiting Race Condition for Wi-Fi Denial of Service Attacks. In *13th International Conference on Security of Information and Networks*. 1–8.

[22] Asier Martínez, Urko Zurutuza, Roberto Uribeetxeberria, Miguel Fernández, Jesus Lizarraga, Ainhoa Serna, and Iñaki Vélez. 2008. Beacon frame spoofing attack detection in IEEE 802.11 networks. In *2008 Third International Conference on Availability, Reliability and Security*. IEEE, 520–525.

[23] Chris McMahon Stone, Tom Chothia, and Joeri de Ruiter. 2018. Extending automated protocol state learning for the 802.11 4-way handshake. In *European Symposium on Research in Computer Security*. Springer, 325–345.

[24] Robert Moskowitz. 2003 (Accessed 26 December 2021). Weakness in passphrase choice in WPA interface. https://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html.

[25] Snehasish Parhi. 2012. Attacks Due to Flaw of Protocols Used In Network Access Control (NAC), Their Solutions and Issues: A Survey. *International Journal of Computer Network and Information Security* 4, 3 (2012), 31.

[26] Konstantinos Pelechrinis, Marios Iliofotou, and Srikanth V Krishnamurthy. 2010. Denial of service attacks in wireless networks: The case of jammers. *IEEE Communications surveys & tutorials* 13, 2 (2010), 245–257.

[27] ESET Experimental Research and Detection Team. 2020 (Accessed 24 October 2021). Kr00k: A serious vulnerability deep inside Wi-Fi encryption. https://www.eset.com/int/kr00k/.

[28] Mark Rison, Mathy Vanhoef, Mark Hamilton, and Jouni Malinen. 2021 (Accessed 27 October 2021). On FrAttacks and related matters. https://mentor.ieee.org/802.11/dcn/21/11-21-1128-00-000m-on-frattacks-and-related-matters.docx.

[29] Domien Schepers, Aanjhan Ranganathan, and Mathy Vanhoef. 2021. Let numbers tell the tale: measuring security trends in wi-fi networks and best practices. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 100–105.

[30] Domien Schepers, Mathy Vanhoef, and Aanjhan Ranganathan. 2021. A framework to test and fuzz wi-fi devices. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 368–370.

[31] Spacehuhn. 2022 (Accessed 4 February 2022). ESP8266 Deauther Version 2. https://github.com/SpacehuhnTech/esp8266_deauther.

[32] Mathy Vanhoef, Prasant Adhikari, and Christina Pöpper. 2020. Protecting wi-fi beacons from outsider forgeries. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 155–160.

[33] 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 *Proceedings of the 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 34–39.

[34] Mathy Vanhoef and Frank Piessens. 2014. Advanced Wi-Fi attacks using commodity hardware. In *Proceedings of the 30th Annual Computer Security Applications Conference*. 256–265.

[35] Mathy Vanhoef and Frank Piessens. 2017. Denial-of-service attacks against the 4-way wi-fi handshake. In *Proceedings of the 9th International Conference on Network and Communications Security*. Academy & Industry Research Collaboration Center (AIRCC), 85–94.

[36] Mathy Vanhoef and Frank Piessens. 2017. Key reinstallation attacks: Forcing nonce reuse in WPA2. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1313–1328.

[37] Mathy Vanhoef and Frank Piessens. 2018. Release the Kraken: new KRACKs in the 802.11 Standard. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 299–314.

[38] Mathy Vanhoef and Eyal Ronen. 2019. Dragonblood: A Security Analysis of WPA3's SAE Handshake. *IACR Cryptol. ePrint Arch.* 2019 (2019), 383.

[39] Mathy Vanhoef, Domien Schepers, and Frank Piessens. 2017. Discovering Logical Vulnerabilities in the Wi-Fi Handshake Using Model-Based Testing. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17)*. Association for Computing Machinery, New York, NY, USA, 360–371. https://doi.org/10.1145/3052973.3053008

[40] Weijia Wang and Haihang Wang. 2011. Weakness in 802.11 w and an improved mechanism on protection of management frame. In *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 1–4.

A IEEE 802.11 STANDARD RULES

Here we list the rules in section 12.6.19 of the IEEE 802.11 standard that specify how stations should handle (unprotected) robust management frames [18, §12.6.19]. We use MFPCap and MFPUnprot as a shorthand for dot11RSNAProtectedManagementFramesActivated and dot11RSNAUnprotectedManagementFramesAllowed, respectively. To make analysis easier, when rules apply conditionally, we prefix the rules between brackets with the conditions that the rule covers. In particular, the prefix *unicast* and *group* specify that a rule covers frames with unicast or group reciever address, respectively. Similarly, the conditions on *MFPCap* and *MFPUnprot* are also included in the prefix where applicable:

(1) *[MFPCap=0, unicast]* A STA with MFPCap equal to false shall transmit and receive unprotected individually addressed robust Management frames to and from any associated STA and shall discard protected individually addressed robust Management frames received from any associated STA.

(2) *[MFPCap=1 station only, MFPUnprot=1, unicast]* A STA with MFPCap equal to true and MFPUnprot equal to true shall transmit and receive unprotected individually addressed robust Management frames to and from any associated STA that advertised MFPC = 0

and shall discard protected individually addressed robust Management frames received from any associated STA that advertised MFPC = 0.

(3) *[MFPCap=1 both, MFPUnprot=1, unicast]* A STA with MFPCap equal to true and MFPUnprot equal to true shall transmit and receive protected individually addressed robust Management frames to and from any associated STA that advertised MFPC = 1, shall discard unprotected individually addressed robust Action frames received from any STA that advertised MFPC = 1, and shall discard unprotected individually addressed Disassociation and Deauthentication frames received from a STA that advertised MFPC = 1 after the PTK and IGTK have been installed. The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK and IGTK are installed.

(4) *[MFPCap=1, MFPUnprot=0, unicast]* A STA with MFPCap equal to true and MFPUnprot equal to false shall transmit and receive protected individually addressed robust Action frames to and from any STA, shall not transmit unprotected individually addressed robust Action frames to any STA, and shall discard unprotected individually addressed robust Action frames received from a STA after the PTK and IGTK have been installed. The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK and IGTK are installed.

(5) *[MFPCap=1 both, group]* A STA with MFPCap equal to true shall discard group addressed robust Management frames received from any associated STA that advertised MFPC = 1 if the frames are unprotected or if a matching IGTK is not available.

(6) *[MFPCap=1, MFPUnprot=0, group]* A STA with MFPCap equal to true and MFPUnprot equal to false shall discard received group addressed robust Management frames that are unprotected or for which a matching IGTK is not available.

(7) *[MFPCap=0, group]* A STA with MFPCap equal to false shall transmit group addressed robust Management frames unprotected and shall ignore the protection on received group addressed robust Management frames.

(8) The STA shall discard any robust Action frames received before the PTK and IGTK are installed.

Although we do not focus on the handling of robust action frames, we do remark that rule (8) stands out as it is not conditional on whether the station supports or requires MFP. In particular, when MFP is not supported, this rule does not appear useful. Another curiosity is rule (4): it mentions the IGTK yet only deals with unicast robust action frames. In other words, there is no need for rule (4) to be partly conditional on a station possessing the IGTK. We consider it interesting future work to further study the handling of robust action frames.