Association Attacks in IEEE 802.11: Exploiting WiFi Usability Features

George Chatzisofroniou(✉) and Panayiotis Kotzanikolaou(✉)

SecLab, Department of Informatics, University of Piraeus, Pireas, Greece
sophron@latthi.com, pkotzani@unipi.gr

Abstract. Association attacks in IEEE 802.11 aim to manipulate wireless clients into associating with a malicious access point, usually by exploiting usability features that are implemented on the network managers of modern operating systems. In this paper we review known association attacks in IEEE 802.11 and we provide a taxonomy to classify them according to the network manager features that each attack exploits. In addition, we analyze the current applicability status of association attacks, by implementing them using the well-known Wifiphisher tool and we review the security posture of modern network managers against known association attacks and their variations. Our results show that association attacks still pose an active threat. In particular, we analyze various strategies that may be implemented by an adversary in order to increase the success rate of association attacks, and we show that even though network managers have hampered the effectiveness of some known attacks (e.g. KARMA), other techniques (e.g. Known Beacons) are still an active threat.

1 Introduction

WiFi, or IEEE 802.11 wireless networking, is probably the most popular type of network for wireless home networking, as well as for network sharing of public or guest networks. Most people expect a standard degree of connectivity wherever they go, while organizations rely on WiFi and other wireless protocols to maintain their productivity. However, since its existence WiFi has been subject to various attacks [5,26,28].

Association attacks are an instance of a man-in-the-middle attacks in WiFi networks. Essentially, they exploit vulnerabilities in the access point selection phase of IEEE 802.11, since the loose definition of this phase leaves room to the vendors for different stack behavior. And since many vendors prioritize usability instead of security, several user-friendly functionalities implemented in most Operating System (OS) network managers, may allow an attacker to fool a client into connecting (associating) with a rogue access point. As vendors are implementing usability features to make the WiFi experience smoother for the end-user, new attacks are keep coming to the surface by exploiting vulnerabilities of the newly added user functionality features.

T. Groß and T. Tryfonas (Eds.): STAST 2019, LNCS 11739, pp. 107–123, 2021.
https://doi.org/10.1007/978-3-030-55958-8_6

Since a wide range of software has inadequate protection against man-in-the-middle attacks, the exposure against such attacks is high. After successfully associating with a victim device, an attacker will be able to intercept part of, or all its network traffic or even leverage this position to exploit device-specific vulnerabilities. Sophisticated phishing attacks may lead to the capture of credentials (e.g. from third party login pages or WPA/WPA2 Pre-Shared Keys) [11]. Note that association attacks may be part of attacks with wider scale. For example, an adversary may be able to expose the real MAC addresses of connected mobile devices bypassing any privacy controls, such as MAC address randomization (e.g. by exploiting Hotspot 2.0 capabilities [9]), and track the location of the victim users.

Several approaches that detect association attacks in IEEE 802.11 have been proposed. Most of them work by collecting the attributes of the nearby networks (including the radio frequency airwaves) and comparing them with a known authorized list. Other user oriented approaches are identifying differences in the number of wireless hops. These techniques are implemented in Wireless Intrusion and Detection Systems (WIDS) that are deployed in Enterprise environments [27] [18]. However, as already stated, WiFi association attacks are not targeting a particular network, but rather on the client devices and the users themselves; hence they can be applied in WiFi environments where WIDS are not available, by exploiting vulnerable usability features implemented at the client side.

Motivation and Contribution. In this paper we provide a thorough analysis and classification of WiFi association attacks. We analyze their differences and examine the situations in which these attacks may be active. We show that although modern network managers are assumed to provide adequate protection against known association attacks, new variations can still be an active threat, mainly due to the prioritization given by OS vendors to the usability, instead to the security features, of network management software. To demonstrate the applicability of such attacks, we have incorporated most of them in Wifiphisher [12], a well-known open source WiFi security testing tool. Finally, we analyze the differences of modern Operating Systems and discuss their exposure to each WiFi association attack.

Paper Structure. The rest of this paper is structured as follows. In Sect. 2 we provide the necessary background information, including both protocol and implementation details, that is necessary for understanding the internals of the various association attacks. In Sect. 3 we provide a taxonomy of known association attacks, based on the usability features that each attack exploits. To the best of our knowledge this is the first taxonomy of IEEE 802.11 association attacks. In Sect. 4 we review the different implementations of network managers across the modern operating systems and we examine how they react to association attacks. Finally Sect. 5 concludes this paper.

2 Background Information: AP Selection Phase and Related Functionality in IEEE 802.11

As explained in Sect. 1 association attacks take advantage of usability features of OS network managers, that usually aim to enhance user-friendliness by automating the access point selection phase. Therefore, in order to understand the origin of such attacks, we will describe the relevant protocol and implementation details. First we review the Access Point Selection process in IEEE 802.11. Then we review the related usability features implemented in the network managers of the most popular OS.

2.1 Access Point Selection in 802.11

In the IEEE 802.11 specification [16], two basic entities are defined. First, there is the station (STA), a device that has the capability to use the 802.11 protocol (e.g. a laptop, a desktop, smart phone or any other WiFi enabled device). Then there is the access point (AP), a network device that allows stations to connect to a wired network. The access point typically connects to a router via a wired connection as a standalone device and then provides wireless connections using the WiFi technology.

The first step for a station to associate successfully with an access point is to populate a list of nearby WiFi networks. This list is called the *Available Networks List* (ANL) and each wireless network stored in it is resembled by its logical name, the *Service Set Identifier* (SSID), along with its encryption type. For each stored network in the ANL, the station also stores the identifier of the access point, the *Basic Service Set Identifier* (BSSID), which is a 48-bit label. If more than one BSSIDs correspond to the same SSID, then the BSSID of the AP with the strongest signal is stored in the ANL.

The station can construct the ANL using two different scanning methods (see Fig. 1): a) *passive scanning*, where the station detects special frames called beacon frames that are periodically sent from the access points to announce the presence of a wireless network, or b) *active scanning*, where the station sends a probe message, called null probe request frame, which asks all access points within the wireless range to respond directly with the necessary information required to establish a connection.

After the ANL is populated (i.e. contains at least one nearby network), the station decides if it will attempt connection to one of the networks stored in it. This decision can be made either automatically, by the OS utilizing usability features discussed in the next section, or manually by the end-user who may select one of the networks in the ANL via a user interface. When a wireless network is selected, the station will pick the associated BSSID from the ANL and send a directed probe request to the corresponding access point to ensure that it is within the area. The access point will respond with a probe response. After this process, both endpoints are ready to proceed to the Authentication and Association Process.

During the Authentication and Association Process, each side needs to prove the knowledge of some credentials. Notably in "Open Authentication", as its name implies, there are no credentials and any wireless device can 'authenticate' (connect) to the access point. Once authentication is complete, mobile devices can finally associate with the access point. The association process will allow to the access point to record each station in order to send and transmit data frames to it.

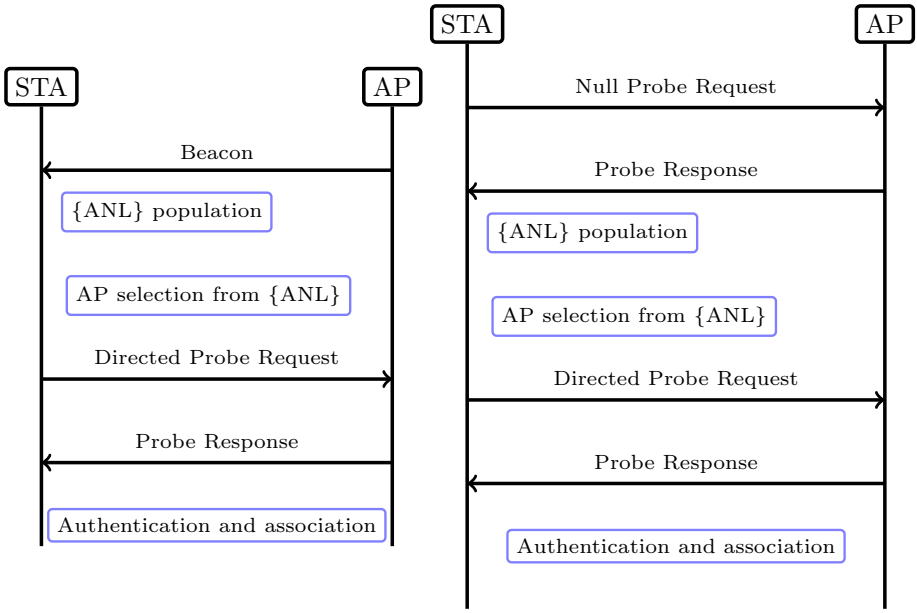

Fig. 1. ANL Population using: (a) Passive Scanning (left) and (b) Active Scanning (right)

2.2 Usability Features Related to AP Selection

Software vendors have introduced a number of features to automate the process of AP selection without requiring user interaction. In this section we explain in detail these features and discuss their underlying logic.

Network Manager Implementation Features. The features that are related with the AP selection phase, usually make use of a special list, called the *Preferred Network List* (PNL). In contrast to the ANL that contains all nearby WiFi networks along with the BSSID of the strongest access point, the PNL contains only networks (SSID and encryption type) that the wireless station will prefer to associate, if they exist within the wireless range. As soon as the ANL is populated and during the AP selection phase, the station will automatically connect to the strongest access point in the intersection of the ANL and the

PNL. If there are no networks around that are also stored in the PNL (i.e. if the intersection of the ANL and the PNL is empty), the station will remain unauthenticated and unassociated until the user manually selects a network.

$$ANL = [wlan_1 : bssid_1, wlan_2 : bssid_2, ..., wlan_n : bssid_n] \quad (1)$$

$$PNL = [wlan_1, wlan_2, ..., wlan_n] \quad (2)$$

$$wlan = [ssid, encryption\ type] \quad (3)$$

The *"auto-reconnect"* feature automatically adds the attributes (SSID and encryption type) of a network to the PNL upon the first connection. The network is usually stored in the PNL until the user manually 'forgets' it. In most operating systems, the default behavior of the auto-reconnect feature differs on the encryption type of the network that the station connects to.

The *"available to all system users"* feature is an extension of the auto-reconnect feature and exists only to multi-user operating systems where each system user has its own version of a PNL. When this feature is enabled, the PNL becomes global across users. For example, if a user adds a network to the PNL, e.g. due to the 'auto-reconnect' feature, then that network will also exist to the PNL of all other users due to the 'available to all users' feature.

The *"active scanning for networks in the PNL"* is another usability feature where the station sends directed probe request frames for networks the stations have associated with in the past (i.e. they exist in the PNL) even if these networks are not around (i.e. they are not in the ANL).

The *"automatically connect to high-quality open networks"* feature allows certain devices to automatically connect to specific high and reliable open networks according to a specific vendor. An example of this feature is the "WiFi Sense" that was introduced by Microsoft in 2016 and it allowed a Windows10 or Windows Phone 8.1 device to automatically connect to suggested open hotspots (WiFi Sense networks). The WiFi Sense feature was removed by Microsoft shortly after the associated risk that we discuss later in this paper was revealed. However, a similar feature was introduced by Google, called WiFi assistant. This feature can be found on Pixel and Nexus devices using Android 5.1 and up in selected countries and it allows automatic connection to open WiFi networks that Google verify as reliable.

Finally, the *"turn on WiFi automatically"* feature will turn the WiFi connection back on when the device is near a network that exists in the PNL of the device.

802.11 Protocol Features: WiFi Roaming and WiFi Direct. According to the WiFi specification, an *Extended Service Set* (ESS) may be formed by deploying multiple access points that are configured with the same SSID and security settings. *WiFi roaming* is an operation where the station decides that is time to drop one AP and move to another (in the same ESS). The operation is completely dependent on the client device; as we have discovered after testing, in

modern operating systems a WiFi client device will typically attempt to maintain a connection with the access point that can provide the strongest signal within a service set.

WiFi Protected Setup Push Button Configuration (WPS-PBC) [3] is an operation where the user presses a (virtual) button on the wireless station and a physical button on the router within 120 s in order for the device to automatically connect to the wireless network without requiring to input any passphrase.

The *WiFi Direct* protocol [10] is built upon the IEEE 802.11 infrastructure and it enables the devices to form P2P groups by negotiating which device will be the Group Owner and which devices will be the clients. WiFi Direct is mainly used for data sharing, video streaming and gaming.

3 Association Attacks: A Taxonomy

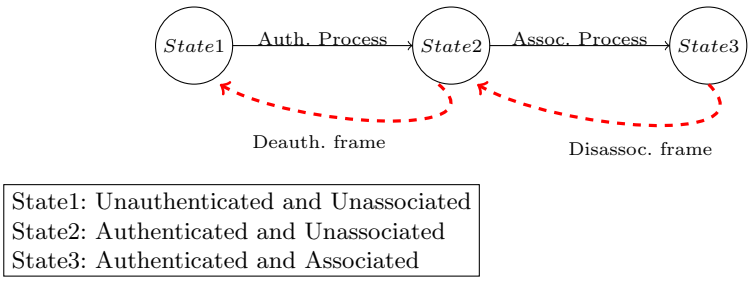

Fig. 2. Dissassociation and deauthentication

As explained above, the goal of association attacks is to trick the target wireless stations into associating to an attacker controlled AP. Association attacks can be categorized based on the feature of the Network Manager that they exploit (Fig. 3). We divide them into two main categories: (a) the *automatic association attacks* where the only prerequisite is that the victim node is within the range of the attacker-controlled AP, which are analyzed in Sect. 3.1; and (b) the attacks that *require user interaction*, which are analyzed in Sect. 3.2. Finally, in Sect. 3.3 we calculate and discuss the exploitability scores of the attacks.

3.1 Automatic Association Attacks

In order to perform an automatic WiFi association attack, the victims stations need to run their Access Point Selection algorithm so that they can be later lured into the rogue network by abusing different features of their network manager.

This can be achieved by traversing the WiFi stations to a state where they are not authenticated nor associated with any AP. In this state, the victim stations will be enforced to run the Access Point Selection algorithm to connect with an

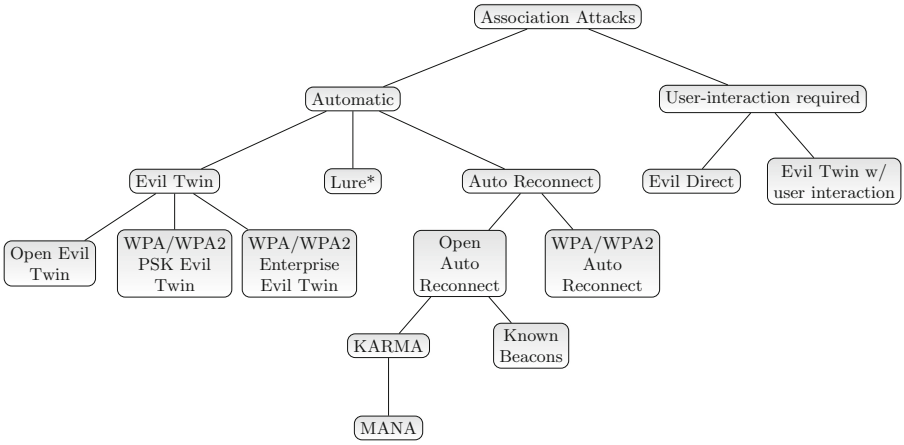

Fig. 3. Classification of WiFi association attacks

AP and maintain connectivity. The most common way to traverse authenticated and associated WiFi stations to an unauthenticated and unassociated state is by forging "Deauthenticate" and "Disassociate" packets as shown in Fig. 2. This can be easily achieved by an attacker, since a known issue with 802.11 is that the management packets are not cryptographically protected against eavesdropping, modification or replay attacks [19]. Alternatively, radio jamming is another common method to block or interfere with authorized wireless communications. An attacker can use an Software Defined Radio (SDR) or cheap off the-shelf WiFi dongles to transmit radio signals and make a wireless channel unusable for other devices [25].

Another way to enforce WiFi stations to run their Access Point Selection algorithm is by enforcing them to restart the WiFi feature itself. This can be done either programmatically (e.g. malware applications within the device may be able to restart the WiFi service) or by abusing the Enable WiFi automatically feature. By broadcasting a WPA/WPA2 network that exists in the PNL of the victim device, it is possible to enforce the device that has this feature enabled, to turn its WiFi feature on and run the Access Point Selection algorithm.

Evil Twin. During an Evil Twin attack [21,29], the adversary copies the ESSID and the encryption type of the wireless network that the victim device is connected to and sets up an AP that broadcasts the same attributes. If the malicious AP offers a stronger signal, and due to the operation of WiFi roaming, the victim device will automatically connect to the rogue network.

The Evil Twin attack is common against public hotspots that are usually employed along with a captive portal mechanism and are commonly deployed in airports, hotels and coffee shops. The adversary can easily replicate both the ESSID and the encryption type (Open) of these networks and assuming that the

rogue AP offers a stronger signal, the victim stations will automatically connect to the malicious network.

The Evil Twin attack is also possible in WPA/WPA2 WiFi networks with known or disclosed Pre-Shared Keys (PSK) or in infrastructures whose members are dynamically joining and leaving the network (e.g. a conference WiFi). In such cases, the secret is either published or known by many parties, thus it can be easily known by a malicious party. Knowing the PSK, the adversary can replicate the ESSID and the encryption of the legitimate Access Point and as in open setups, the clients of these networks will automatically connect to the rogue Access Point. Finally, the Evil Twin attack is also popular against Enterprise networks [6,20] that are widely used in large corporations. If the victim stations are not validating the server certificate presented by the AP, the corporate setups are vulnerable to Evil Twin.

Attack Requirements:

- Victim device is connected to a wireless network
- Physical position within the Wi-Fi range of the victim device
- Knowledge of the wireless network's SSID that the victim device is connected
- In case the victim is connected to WPA/WPA2 network, knowledge of it secret (e.g. PSK)

Attack Steps:

1. Obtain position within the Wi-Fi range of the network that the victim device is connected
2. Spawn a network with the same SSID and encryption type as the network that the victim device is connected. Rogue network's signal strength needs to be stronger than the legitimate.

Lure*. The Lure*-type attacks are abusing the "automatically connect to high-quality open networks" that is found in some Operating Systems. The first attack of this type was Lure10 [13] that used to exploit the WiFi Sense feature found on Windows 10 and Widows Phone 8.1. This attack relied on the victim's device being fooled into believing it is within the geographical area of a WiFi Sense-tagged open wireless network. This could be achieved by broadcasting beacon frames of that area and eventually tricking the Windows Location Service [23]. Finally the attacker would successfully mimic the WiFi Sense network in that area (broadcasting the same SSID was found to be enough) and the victim users would connect to the rogue AP.

While the Lure10 attack is no longer applicable since the removal of the WiFi Sense feature by Microsoft, a similar attack vector recently appeared on certain Android devices with the introduction of the Google Assistant, which also enables the automatic connection to open networks.

Attack Requirements:

- Physical position within the Wi-Fi range of the victim device
- A feature that allows automatic connection to vendor-suggested hotspots is enabled on the victim's device
- Knowledge of a vendor-suggested hotspot's SSID
- Requirements for Location Service or GPS spoofing should also be satisfied

Attack Steps:

1. Obtain position within the Wi-Fi range of the victim device
2. If needed, traverse the WiFi station to a state where it is not authenticated nor associated with any AP
3. Spoof GPS or Location Service in order to "transfer" the victim in the vendor-suggested hotspot's location
4. Spawn a rogue network with the same SSID as the vendor-suggested network

Auto Reconnect Exploitation. In order to exploit the auto-reconnect feature, the adversary will spawn a network that is stored in the PNL of the target device. In contrast to the Evil Twin where the rogue network is copied based on the network that the target device is currently connected to, in this attack the rogue network is *not* required to exist in the wireless range of the victim device. If the victim's device is not authenticated to any network, it will automatically join the rogue network that was spawned by the attacker, regardless the fact that the attributes of this network may have been added to the PNL a long time ago in a complete different environment.

In the Open Auto Reconnect scenario, the attacker only needs to replicate the SSID of the network that exists in the PNL, while in the case of WPA/WPA2 encryption, the Pre-Shared Key (PSK) is also required. If the PSK of a WPA/WPA2 network is not known, an attacker may leverage public crowdsourced databases to retrieve the secret and successfully mimic the network that exists in the victim's PNL. This attack typically requires some familiarity with the victim user and his whereabouts in order for the adversary to guess the attributes of a network in the target device's PNL. If the "available to all system users" flag is enabled, the attack surface is increased; networks that were stored as part of the association process of other users in the target system can be leveraged to carry out the attack.

Even if the whereabouts of the victim user are not known, an attacker that has achieved local access to the remote station (e.g. by infecting the victim device with a malware) will be able to add a network to the PNL of that host that can be leveraged later to carry out the attack. These kind of "backdoor" networks may also be added to the victim stations by physical means. Notably, in a host running Windows 10, even if the workstation is locked, an adversary with physical access may still connect to a wireless network that will be eventually added to the PNL of this device [24].

Attack Requirements:

- Physical position within the Wi-Fi range of the victim device
- Auto-Reconnect flag is enabled on victim's device
- Knowledge of an unencrypted wireless network's SSID that exists in the victim's PNL
- In case of WPA/WPA2 network, knowledge of the secret (e.g. PSK)

Attack Steps:

1. Obtain position within the Wi-Fi range of the victim device
2. If needed, traverse the WiFi station to a state where it is not authenticated nor associated with any AP
3. Spawn a wireless network with the same SSID as the unencryted wireless network that exists in the victim's PNL

KARMA and MANA. While Open Auto Reconnect attacks exploits the "auto-reconnect" feature, the KARMA attack [15] also exploits the active scanning for networks that stations have associated with in the past. In this attack, a rogue AP is introduced that masquerades as a public network that nearby WiFi clients are actively searching for. Victim stations that are actively looking for open networks stored in their PNL will automatically join the rogue AP.

MANA [22] is an attack that took KARMA a step further by configuring a rogue AP that not only replies to directed probes, but additionally it responds to the victim device's broadcast probe requests (e.g. using the same response). Furthermore, a "loud" mode was introduced where the rogue AP is responding to each device's probe request frames with a list of networks that have been searched for by other devices within the range of the rogue AP.

Attack Requirements:

- Physical position within the Wi-Fi range of the victim device
- Auto-Reconnect flag is enabled on victim's device
- The victim device performs active scanning for networks stored in its PNL
- At least one unencrypted wireless network exists in victim's PNL

Attack Steps:

1. Obtain position within the Wi-Fi range of the victim device
2. If needed, traverse the WiFi station to a state where it is not authenticated nor associated with any AP
3. Respond positively to directed probe requests that are intended for unencrypted networks
4. Additionally, respond to broadcast probe requests using the same response

Known Beacons. The Known Beacons attack [8,14] is also a special instance of an Open Auto Reconnect attack, which is usually applied when the attacker has no prior knowledge of the victims' PNL and is applicable against all modern operating systems. In an attempt to guess the SSID of an open network that exists in the victim device's Preferred Network List, the attacker will broadcast

dozens of beacon frames from a "dictionary" of common SSIDs. The dictionary includes entries with popular SSIDs that are commonly used by network administrators (e.g. 'wireless', 'guest', 'cafe', 'public'), SSIDs of global WiFi networks (e.g. 'xfinitywifi', 'attwifi', 'eduroam', 'BTFON'), SSIDs of hotspots that exist in hotels, airports and other places of public interest (e.g. 'hhonors_public', 'walmartwifi'). Finally, location-specific SSIDs based on the victim users whereabouts can be collected with wardriving [17] or by looking at public databases of 802.11 wireless networks.

Attack Requirements:

- Physical position within the Wi-Fi range of the victim device
- Auto-Reconnect flag is enabled on victim's device
- There is at least one wireless network from the victim's PNL in the dictionary of popular SSIDs

Attack steps:

1. Obtain position within the Wi-Fi range of the victim device
2. If needed, traverse the WiFi station to a state where it is not authenticated nor associated with any AP
3. Broadcast dozens of beacon frames from a dictionary of common SSIDs

3.2 Association Attacks Requiring Interaction

In contrast to the previous category where the attacks can be launched solely at the will of the attacker, in this case the attacks require some user interaction by the victim user (or a victim user initiated process). For this reason, their estimated risk is usually lower. However, these attacks are applicable in cases where the requirements for automatic association attacks are not satisfied.

EvilDirect Attack. The WiFi Direct protocol defines a Group Owner (GO) to allow other clients to connect with. EviDirect attacks [7] the WiFi Direct protocol by spawning a rogue GO that operates on the same channel as the original and has the same MAC address and SSID. If the rogue GO accepts any invitation requests faster than the legitimate one, the adversary will be able to hijack the wireless communications.

The fundamental problem with EvilDirect lies in the underlying WiFi Protected Setup Push Button Configuration (WPS-PBC) protocol which is susceptible to an active attack where the attacker offers an AP in the PBC state on another channel to induce an Enrollee to connect to the rogue network. These techniques require the victim users to actively use the WPS-PBC and WiFi Direct functionalities. Notably, we discovered that this attack is more viable on Windows10 where the WPS-PBC virtual button is automatically pushed just by selecting a network with WPS capabilities on the networks manager's list and without the end-user's explicit consent.

Attack Requirements:

- Physical position within the Wi-Fi range of the victim device
- Victim user initiates a WPS-PBC request

Attack Steps:

1. Obtain position within the Wi-Fi range of the victim device
2. Wait until the victim user activates WiFi Direct on the device
3. Accept the invitation request faster than the legitimate GO

Evil Twin (Requiring User Interaction). As in the case of the automatic Evil Twin, this attack is also based on the replication of a legitimate AP, however it requires some user interaction. The replicated rogue Access Points have at least one of their attributes (i.e. SSID and encryption type) different from the legitimate AP. In our experience, this may happen for two reasons: In the first case, the adversary cannot replicate the encryption type of the legitimate AP (e.g. because the PSK is unknown). In this scenario, the adversary will commonly perform a downgrade attack by spawning an Open-type network. Interestingly, from our research, it appears that only macOS systems will issue a warning for downgrade attacks.

In the second case, the adversary targets an Open-type network in an infrastructure where new members are dynamically joining the network (e.g. in public areas). In this scenario, it is reasonable for the attacker to spawn a rogue network with an SSID that precedes alphabetically from the target's AP SSID. Since network managers order the networks of the same signal power in an alphabetic order, the adversary raises the chances of having the rogue AP shown first in the network manager's list, hence victim users are more likely to select it. The attacker can take this a step further by spawning intermediate networks (i.e. by mounting an SSID flooding attack) in an attempt to push the legitimate SSID further down the Network Manager's list.

Attack Requirements:

- Physical position within the Wi-Fi range of the victim device
- Victim user is fooled into choosing to connect to the rogue Access Point

Attack Steps:

1. Obtain position within the Wi-Fi range of the victim device
2. Spawn a rogue Access Point that has at least one of its attributes different from the legitimate AP
3. Fool victim user into selecting the rogue Access Point

3.3 Association Attacks Exploitability

We used the exploitability sub-score equation that exists in CVSS 3.1 [4] to calculate the exploitability scores that reflect the ease and technical means by which each association attack can be carried out. We assumed an attacker that is positioned within the Wi-Fi range of an area with a moderate number of users

Table 1. Exploitability matrix of association attacks

Association attack	Exploitability metrics				Exploitability score (0–3.9)
	Attack vector	Attack complexity	Privileges required	User interaction	
Open Evil Twin	Network	High	None	None	2.2
WPA/WPA2 PSK Evil Twin	Network	High	None	None	2.2
WPA/WPA2 Enterprise Evil Twin	Network	High	None	None	2.2
Lure*	Network	High	None	None	2.2
Auto-Reconnect Open	Network	High	None	None	2.2
Auto-Reconnect WPA/WPA2	Network	High	None	None	2.2
Known Beacons	Network	Low	None	None	3.9
KARMA	Network	Low	None	None	3.9
MANA	Network	Low	None	None	3.9
EvilDirect	Network	Low	None	Required	2.8
Evil Twin w/ user interaction	Network	High	None	Required	1.6

(e.g. 50–100 devices). Finally, we considered that an attack is successful if at least one device is associated with the attacker-controlled AP.

In Table 1 we outline all association attacks with their exploitability metrics and the calculated scores. It is notable that KARMA, MANA and Known Beacons attacks have the higher exploitability score due to their automatic nature and their low complexity. The attack with the lowest exploitability score is "Evil Twin w/ user interaction" because of the required user interaction and the difficulty of the conditions that need to be satisfied to mount the attack.

4 Analysis of Network Managers' Behavior

4.1 Attack Implementation

We implemented Evil Twin and Auto-Reconnect attacks against 802.11 clients using Python standard library modules. We included them in the first release of Wifiphisher that was published under GPLv3 [2]. The rest of the association attacks and de-authentication techniques, were implemented as "ifiphisher extensions" which are scripts in Python that are executed in parallel and expand the functionality of the main Wifiphisher engine. For time-critical operations we developed "roguehostapd" [1], a fork of hostapd, that communicates with the main Wifiphisher engine by providing Python bindings with ctypes.

Running Wifiphisher requires at least one wireless network adapter that supports AP and Monitor mode in order to sniff and inject wireless frames. Wi-Fi drivers should also support the Netlink socket family.

Table 2. Usability features on modern Operating Systems

Operating System	Auto-reconnect		Avail. to all system users	Probes for prev. conn. networks	Auto-enable WiFi	Auto-connect to high-quality WiFi networks	Connect to network with locked screen
	Open	WPA/WPA2					
Windows10	✓[1]	✓	✓				✓
macOS	✓	✓	✓				
Android	✓	✓			✓	✓[1]	
iOS	✓	✓					

Comments	(1) The feature is available but is disabled by default

4.2 Result Analysis

We examined the behavior of modern Operating Systems against known association attacks that were described in the previous sections of this paper. Specifically, in desktop systems, we analyzed the behavior of Windows10 and macOS 10.15, while in mobile devices, we examined Android 9 and iOS 12.4.

In Table 2 we summarize all existing usability features that are supported by the examined Operating Systems, and we also identify which features are enabled by default. The dissimilarities are notable. It can be concluded that each OS was designed with a different threat model in mind given that the risk involved with these usability features is known for some time now. For example, Windows10 will not allow automatic connection to previously connected open networks by default. However, the vendor seems to accept the risk of a physical attacker adding a network to the PNL (i.e. adding a network with locked screen is enabled) while the rest of the OS show the exact opposite behavior.

Mobile devices appear to have more usability features enabled by default than desktop operating systems. We find this reasonable since mobile devices rely on both user-owned and externally-managed WiFi connectivity.

It also seems that most of the vendors have stopped the probes to previously connected networks in order to hamper the effectiveness of KARMA and MANA attacks. However, they do accept the risk involved with leaving the Auto-reconnect feature enabled that makes them susceptible to Known Beacons.

In Table 3 we outline all association attacks and we show the Operating Systems that are vulnerable to each one of them. We can conclude that even though network managers have removed some of the risky features (for example those related with the KARMA attack), other association attacks are still active. Known Beacons appears to be the most effective WiFi association attack against modern Operating Systems. It is also worth mentioning that in a real scenario and depending on the identified vulnerabilities/effective usability features, an attacker will use a combination of the above attacks, for example KARMA and Known Beacons at the same time.

Table 3. Current landscape of association attacks

Association Attack	Exploitability Score	Exploited Usability Features					Vulnerable Operating Systems			
		Auto Reconnect		Avail. to all system users	Probes for prev. conn. networks	Other	Windows10	macOS	Android	iOS
		Open	WPA/WPA2							
Open Evil Twin	2.2						✓	✓	✓	✓
WPA/WPA2 PSK Evil Twin	2.2						✓	✓	✓	✓
WPA/WPA2 Enterprise Evil Twin	2.2						✓	✓	✓	✓
Lure*	2.2					✓[2]			✓[5]	
Auto-reconnect Open	2.2	✓		✓[1]			✓[4]	✓	✓	✓
Auto-reconnect WPA/WPA2	2.2		✓	✓[1]			✓	✓	✓	✓
Known Beacons	3.9	✓		✓[1]			✓[4]	✓	✓	✓
KARMA	3.9	✓		✓[1]	✓					
MANA	3.9	✓			✓					
EvilDirect	2.8					✓[3]	✓	✓	✓	✓
Evil Twin /w user interaction	1.6						✓	✓	✓	✓

Comments	(1) The feature increases the success rates but is not required for the attack to be successful (2) Automatically connect to high-quality open networks (3) WiFi Direct (4) Not vulnerable by default (5) Specific versions only

5 Conclusions

Since 802.11 leaves room for custom implementations regarding the WiFi association phase, Operating System vendors tend to prioritize usability features instead of security. In this paper we have analyzed how these usability features can be exploited by various WiFi association attacks and we have validated the behavior of modern OS network managers, by implementing these attacks using Wifiphisher. Users that want to protect themselves from automatic association attacks need to disable the relevant features and revoke the Wi-Fi permission for all installed applications. Using a VPN solution right after associating with an access point is also an effective countermeasure assuming that the VPN client properly authenticates the other endpoint. As a future work, we plan to extend our analysis in other WiFi protocol features and to propose protocol extensions that will provide adequate security against WiFi association attacks.

Acknowledgement. This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH-CREATE-INNOVATE (project code: T1EDK-01958).

This work has been partly supported by the University of Piraeus Research Center.

References

1. Roguehostapd github page. https://github.com/wifiphisher/roguehostapd
2. Wifiphisher github page. https://github.com/wifiphisher/wifiphisher
3. Wi-fi protected setup specification version 1.0h. 2006 (2015)
4. Common vulnerability scoring system version 3.1: Specification document (2019). https://www.first.org/cvss/specification-document
5. Pwning WiFi networks with bettercap and the PMKID client less attack, February 2019. https://www.evilsocket.net/2019/02/13/Pwning-WiFi-networks-with-bettercap-and-the-PMKID-client-less-attack/
6. Cassola, A., Robertson, W., Kirda, E., Noubir, G.: A practical, targeted, and stealthy attack against WPA enterprise authentication. In: NDSS Symposium 2013, June 2013. https://doi.org/10.1109/IAW.2005.1495975
7. Altaweel, A., Stoleru, R., Gu, G.: EvilDirect: A new Wi-Fi direct hijacking attack and countermeasures. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN), pp. 1–11, July 2017. https://doi.org/10.1109/ICCCN.2017.8038416
8. Dagelić, A., Perković, T., Vujatović, B., Čagalj, M.: SSID oracle attack on undisclosed Wi-Fi preferred network lists. Wirel. Commun. Mob. Comput. **2018**, 15 p. (2018). https://doi.org/10.1155/2018/5153265. Article ID 5153265
9. Barbera, M.V., Epasto, A., Mei, A., Perta, V.C., Stefa, J.: Signals from the crowd: uncovering social relationships through smartphone probes. In: Proceedings of the 2013 Conference on Internet Measurement Conference, pp. 265–276. ACM (2013)
10. Camps-Mur, D., Garcia-Saavedra, A., Serrano, P.: Device-to-device communications with Wi-Fi direct: overview and experimentation. IEEE Wirel. Commun. **20**(3), 96–104 (2013). https://doi.org/10.1109/MWC.2013.6549288
11. Chatzisofroniou, G.: Efficient Wi-Fi phishing attacks. Tripwire blog (2017)
12. Chatzisofroniou, G.: Introducing wifiphisher. In: BSidesLondon 2015 (2017)
13. Chatzisofroniou, G.: Lure10: Exploiting windows automatic wireless association algorithm. In: HITBSecConf 2017 (2017)
14. Chatzisofroniou, G.: Known beacons attack. CENSUS S.A. blog (2018)
15. Dai Zovi, D.A., Macaulay, S.A.: Attacking automatic wireless network selection. In: Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, pp. 365–372, June 2005. https://doi.org/10.1109/IAW.2005.1495975
16. Group, I.W.: Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: higher-speed physical layer in the 5 GHZ band. In: IEEE Std 802.11 (1999). https://ci.nii.ac.jp/naid/10011815988/en/
17. Hurley, C.: WarDriving: Drive, Detect, Defend: A Guide to Wireless Security (2004)
18. Jana, S., Kasera, S.K.: On fast and accurate detection of unauthorized wireless access points using clock skews. IEEE Trans. Mob. Comput. **9**(3), 449–462 (2010). https://doi.org/10.1109/TMC.2009.145
19. Nobles, P.: Vulnerability of IEEE802.11 WLANs to MAC layer dos attacks. In: IET Conference Proceedings, pp. 14–14(1), January 2004. https://digital-library.theiet.org/content/conferences/10.1049/ic.2004.0670
20. Nussel, L.: The evil twin problem with WPA2-enterprise. SUSE Linux Products GmbH (2010)
21. Roth, V., Polak, W., Rieffel, E., Turner, T.: Simple and effective defense against evil twin access points. In: Proceedings of the First ACM Conference on Wireless Network Security, pp. 220–235. ACM (2008)
22. SensePost: Manna from heaven. DEF CON 22 (2015)

23. Tippenhauer, N.O., Rasmussen, K.B., Pöpper, C., Capkun, S.: iPhone and iPod location spoofing: Attacks on public WLAN-based positioning systems. Technical report/ETH Zürich, Department of Computer Science 599 (2012)
24. Vanhoef, M.: Windows 10 lock screen: abusing the network UI for backdoors (and how to disable it). Mathy Vanhoef blog (2017)
25. Vanhoef, M., Piessens, F.: Advanced Wi-Fi attacks using commodity hardware. In: Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, pp. 256–265. ACM, New York (2014). https://doi.org/10.1145/2664243.2664260
26. Vanhoef, M., Piessens, F.: Key reinstallation attacks: forcing nonce reuse in WPA2. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pp. 1313–1328. ACM, New York (2017). https://doi.org/10.1145/3133956.3134027
27. Venkataraman, A., Beyah, R.: Rogue access point detection using innate characteristics of the 802.11 MAC. In: Chen, Y., Dimitriou, T.D., Zhou, J. (eds.) SecureComm 2009. LNICST, vol. 19, pp. 394–416. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05284-2_23
28. Viehbck, S.: Wi-Fi protected setup online pin brute force vulnerability (2011)
29. Yang, C., Song, Y., Gu, G.: Active user-side evil twin access point detection using statistical techniques. IEEE Trans. Inf. Forensics Secur. **7**(5), 1638–1651 (2012)