Cybersecurity Daily Dairy

Day 8: Wi-Fi Deauthentication & WPA/WPA2 Password Cracking

Date: June 26, 2025

Topics Covered:

- Capturing WPA/WPA2 handshakes
- Wi-Fi deauthentication attacks
- Password cracking using Aircrack-ng and Hashcat
- Monitor mode configuration
- Ethical considerations for wireless testing

What I Did:

Today, I performed a **Wi-Fi deauthentication attack** to disconnect users from a wireless access point. This allowed me to capture the **WPA/WPA2 handshake** when devices attempted to reconnect. Once the handshake was saved, I attempted to **crack the Wi-Fi password** using both dictionary-based and brute-force methods with tools like **Aircrack-ng** and **Hashcat**.

Prerequisites:

- A Linux-based system such as Kali Linux or Arch
- Tools pre-installed:
  - Aircrack-ng suite (includes airodump-ng, aireplay-ng, airmon-ng)
  - Hashcat (for optional brute-force attack)
- A Wi-Fi adapter that supports monitor mode and packet injection

---

Steps Followed:

1. Set Up Monitor Mode

I first identified the name of the wireless interface and enabled **monitor mode**, which allows the adapter to listen to all wireless traffic. Conflicting services were disabled to avoid interruptions.

2. Scan for Target Networks

I scanned for nearby Wi-Fi networks and identified a target by noting its **BSSID** (MAC address) and **channel number**.

3. Capture the WPA/WPA2 Handshake

I locked onto the specific network using its BSSID and channel, and began monitoring traffic while saving the output to a capture file.

4. Perform Deauthentication Attack

To trigger clients to reconnect and generate a handshake, I sent deauthentication frames to either:

- All devices on the network (broadcast), or
- A specific client on the network (targeted).

Once a client reconnected, the WPA handshake was captured and shown in the monitoring interface.

5. Crack the Captured Password

After capturing the handshake:

- I used **Aircrack-ng** with a **dictionary file** (like rockyou.txt) to attempt to crack the password.
- Optionally, I converted the capture file to **Hashcat format** and launched a brute-force attack using Hashcat.

Key Learnings:

- Deauthentication is a powerful technique for handshake capture in WPA/WPA2 networks
- Capturing traffic requires enabling monitor mode and disabling network managers
- The effectiveness of cracking attempts largely depends on the **quality and relevance of the wordlist**
- **Ethical boundaries** must be respected—these techniques should be applied only on authorized networks

---

Important Notes:

- Always perform wireless testing in legal and controlled environments
- Wordlists such as rockyou.txt or customized dictionaries improve the chances of success
- WPA3 is more secure and less vulnerable to this type of attack