WPA2 HACKING WITH RASPBERRY PI

ANDRÉ JESUS
(CYBERSECURITY BOOTCAMP)
June 2, 2024

The Internet of things (IOT's) as we know today is an atmosphere full of constant change and challenges. Information travels from and to all sorts of devices, servers and even Virtual machines. In between all that what seems to be a chaotic Opera there is the common User, the Hacker and someone that acts as an ethical Conductor. However, this is not a Western on *"The Good, the Bad, and the Ugly*", we have on one side of the spectrum Systems that are meant to be broken, while data is meant to be kept confidential and available at the same time.

For this particular project I decided to explore a common attack vector in the modern state of the "Internet", often overlooked: cracking Wireless passwords. As it opens the door to malicious actors to access sensitive data, network admin credentials, backdoor persistence, DDos Attacks and much more.

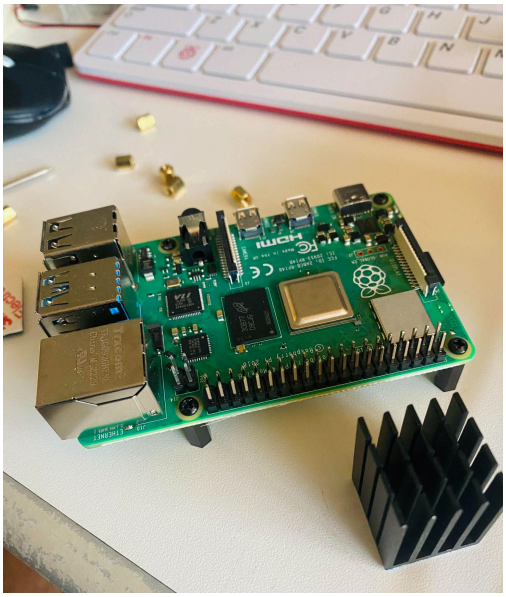
In general it is safe to say that user accounts passwords have indeed become more complex over the past few years, not all but more of them, with the security mindset implemented by Password Managers (From Google for instance). However this idea is often overlooked when users are setting Router credentials, often leaving everything set to default values, and Access Point passwords tend to be simpler, to allow easy connection and sharing. On the other side of the spectrum, even with a more complex password, there are always certain flaws within the Router capabilities subject to exploitation.

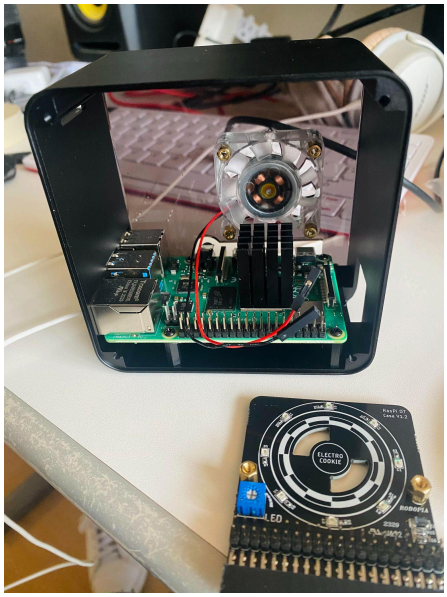
Understanding the outdated vulnerabilities of WEP (Wired Equivalent Privacy), a Wi-Fi security algorithm standardized in the late 90s, I shifted my focus to exploring the vulnerabilities and of its successors: WPA (Wi-Fi Protected Access) and then WPA2, introduced 20 years ago and still widely used today.

Topics

Raspberry Pi 4 Model b (4GB)

In the dynamic field of cybersecurity, the Raspberry Pi stands out as a remarkably versatile and affordable tool. From penetration testing to network monitoring, it can be configured for various security tasks, supported by a community that continually expands its capabilities through open-source contributions. This mini-computer exemplifies how innovation can thrive in compact packages. And this specific model B has a built-in Wifi adapter that supports Monitor mode (Dual Band 2.4Ghz + 5Ghz), but also I can turn it into “Master Mode” to use it as a Rogue Access Point to collect credentials. I then installed a heat sink and fan to dissipate heat, and a cover to keep it secure with Kali Linux Installed on a Micro SD card. For the purposes of this Project, which is to try various wifi hacking programs and scripts I kept it simpler. *





* External WiFi adapters can significantly boost strength and speed. **Alfa** antennas, renowned for their high-gain performance, seamlessly integrate with Kali Linux, making them ideal for robust signal amplification. For more user-friendly GUI interaction advanced capabilities, the **WiFi Pineapple Mark VII** stands out as a top-tier choice in the current market, offering substantial power and versatility for ethical hacking tasks. (No pringles cans needed here =)) It’s also good practice to Use **fail2ban** (for instance) to make your system secure before any monitoring out there..

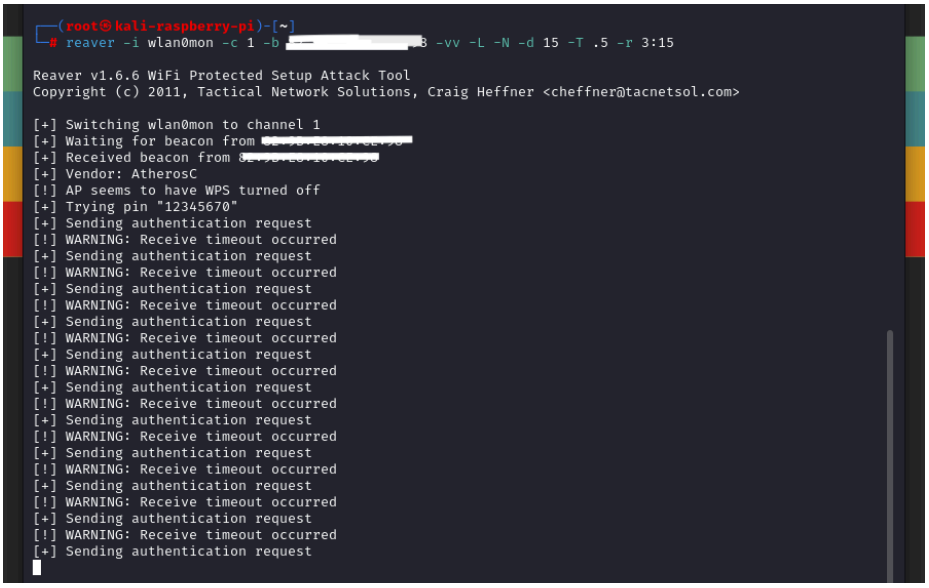
Using commands like airodump-ng wlan0mon and wash -i wlan0mon, I was able to discover various networks within the range of my WiFi monitor. For ethical and educational purposes, I focus solely on analyzing my own router and network. These commands reveal crucial details such as the router's BSSID (Basic Service Set Identifier), operating channel, connected devices, WPS versions, encryption type, and signal strength. It's important to note that WiFi adapters have regulated transmission power, but this can be adjusted to reach more networks. However, it's essential to comply with each country's specific legal guidelines regarding transmission power.

WPS Pin Attack with Reaver

One of the many vulnerabilities of WPA2* is associated with Routers that have WPS (Wi-fi Protected Setup) activated. Which is meant to look for compatible devices within the range and allow them to connect without having to type the Wi-fi long password. This is a problem. As it opens the door for some application reverse engineering. Upon exploring WPS, the way it functions, we know that it uses a 8 digit Pin (the last one being a checksum can be ignored), potentially having a 10^7 (10.000.000 combinations). However, because each half of the Pin is checked independently, we are dealing with much smaller permutations of 10^4 (10.000 combinations) for the first half, and 10^3(1.000 combinations) for the second half. Making it vulnerable to brute force attacks.

Although I was unable to expose the Pin with Reaver, after several hours of persistence and even a few extra command options. This Pin Attack was mostly effective against WPS version 1.0. New Routers carry WPS 2.0. It's also important to note that some Routers are more vulnerable than others to this specific attack, as it depends largely on how the Router is assigning random numbers for the Pin. The vulnerability falls onto the fact that some of those numbers are simple derivations of hashes. For project purposes I was using the Hitron model: Coda 4582 U. WPS 2.0 . Reaver however is still a tool that can be used today on older Routers (using older versions of WPS) but also for Brute Forcing (Pixie Dust Attack). Depending on computing power, If the router is Vulnerable, Reaver can recover the Pin within minutes, despite WPS version.

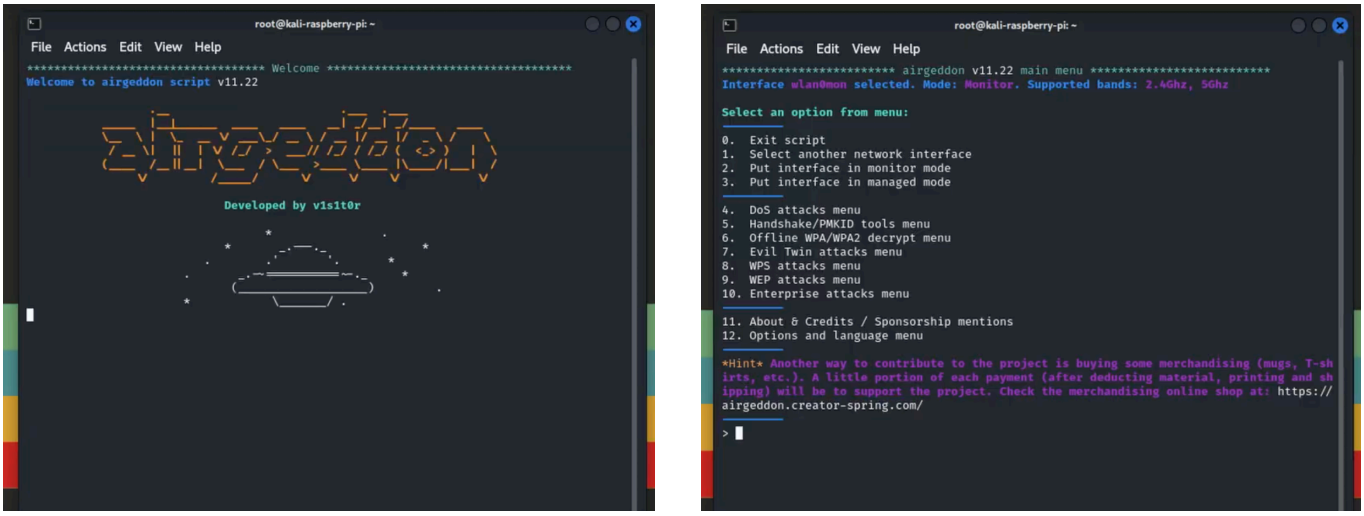
* Please note that WPS is not compatible with WPA2 Enterprise (Meant to be used in Business Environments) , for this project I focus on my own personal network that carries WPA2 Personal.



Reaver: Failed WPS Pin Attack

Pixie Dust + Evil Twin Attack with Airgeddon.

An important aspect of Wi-Fi security that often goes overlooked—WPS, Even with strong and complex Wi-Fi passwords in place, routers can still be vulnerable if WPS is enabled. This vulnerability stems from a brute force technique known as the Pixie Dust Attack. Airgeddon compiles many 3rd party wireless audit tools on a python script framework with a fancy step-by-step menu far from only focusing on WPS attacks. Which is only one attack vector of many. Asides from Reaver, it does allow you to choose Bully to perform a Pixie Dust attack, especially on routers that don't have WPS on "Locked" mode. However after a few minutes it was clear to me that WPS attacks are something that have been patched and secure at least for my specific Router on both Networks for this test.There are many other capabilities worth exploring with this script.



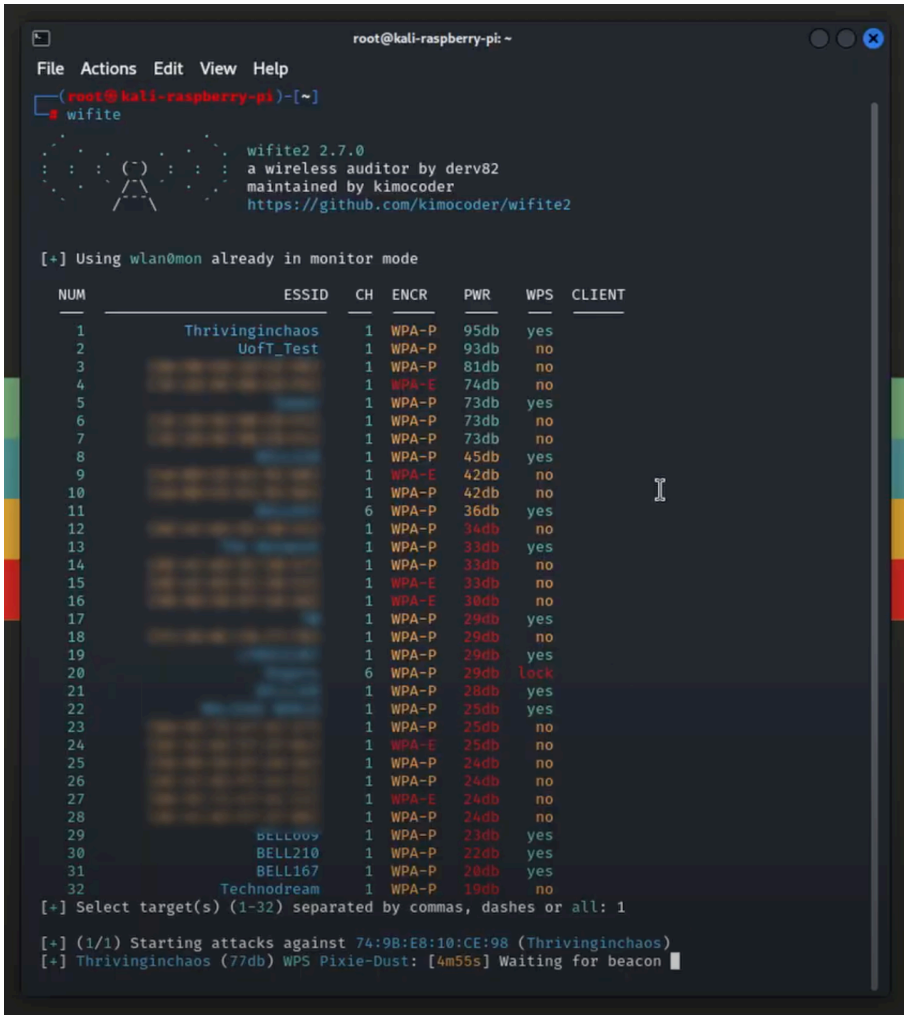
Airgeddon: Versatile attack vectors

Option 7, the Evil Twin attack, begins by capturing devices' Probe Requests for specific WiFi networks (SSIDs). Airgeddon then creates a fake Access Point (AP) with the same SSID. Devices connect to this fake AP, allowing Airgeddon to intercept and manipulate their internet traffic. This attack can capture credentials, execute man-in-the-middle attacks, and intercept WPA/WPA2 handshakes for offline cracking.

A notable feature is the "Evil Twin Attack with Captive Portal." This deceives users into thinking their WiFi is malfunctioning. The captive portal prompts victims to re-enter their passwords to reconnect and undergo a fake update. If the correct password is entered, the hacker can view it in their terminal. Once captured, the portal disappears, leaving the victim unaware of the compromise. This exploit exploits trust and confusion to steal sensitive information from unsuspecting users.

Deauth + Dictionary Attack (Wifite + Aircrack-ng)

Onto exploring different tools I started by running an automated wireless auditor (Wifite) to expose Networks ESSID, their respective channels and information on WPS status. For the purpose of this project I used my own home Wireless internet “ThrivingChaos” (WPS enabled) and also my guest Wi-fi: Uoft_Test (no WPS) for comparison. Wifite runs a few specific automated Attacks including Pixie-Dust, brute force Pin Attack and WPA Handshake Capture (based on Deauthentication method). I had a device disconnecting and connecting from both Wi-fi, to simulate what would happen when someone connects to it with any device in a real world environment.

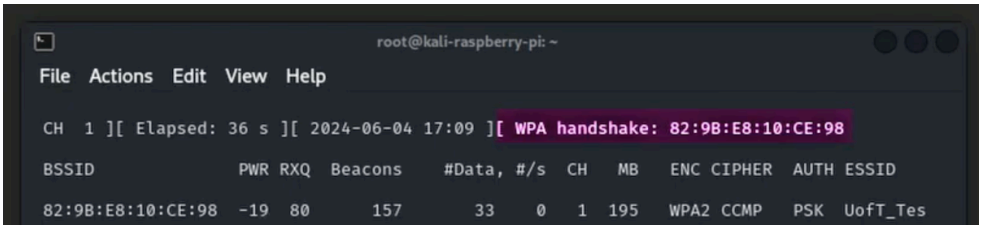


Wifite Auditor, Note that Some Routers have WPS Locked for extra security.

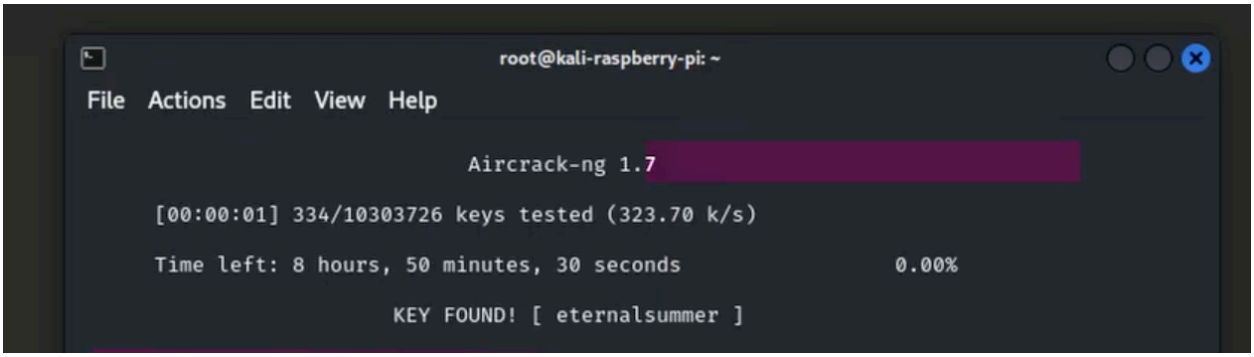
The deauthentication method (as a denial of service attack) relies on “listening” to a specific Network using a IEEE 802.11 wireless frame. Waiting for a device to connect to then send deauth commands, making the user have to reconnect again. The main objective is to capture the WPA Handshake onto a .cap file (that can later be used for a wordlist attack) or to collect more information with tools like Wireshark. Wifite uses worlist-probable.txt as default which wasn’t able to crack the password. However the purpose of this attack was also to capture the WPA Handshake, from there, any other wordlist that had that password, as there are a vast high volume of them available, one of them would definitely crack it using Aircrack-ng and the respective .cap file. Wish takes us to the process of building your own wordlists. In the end it all comes down to capture that first Handshake between a Client and the Router. Which simply can’t be stopped.

Left: Capturing traffic from a specific Network, on to a .cap file. | Right: Deauthenticating specific device connected to the Network.

.



Airodump-ng command, capturing WPA handshake



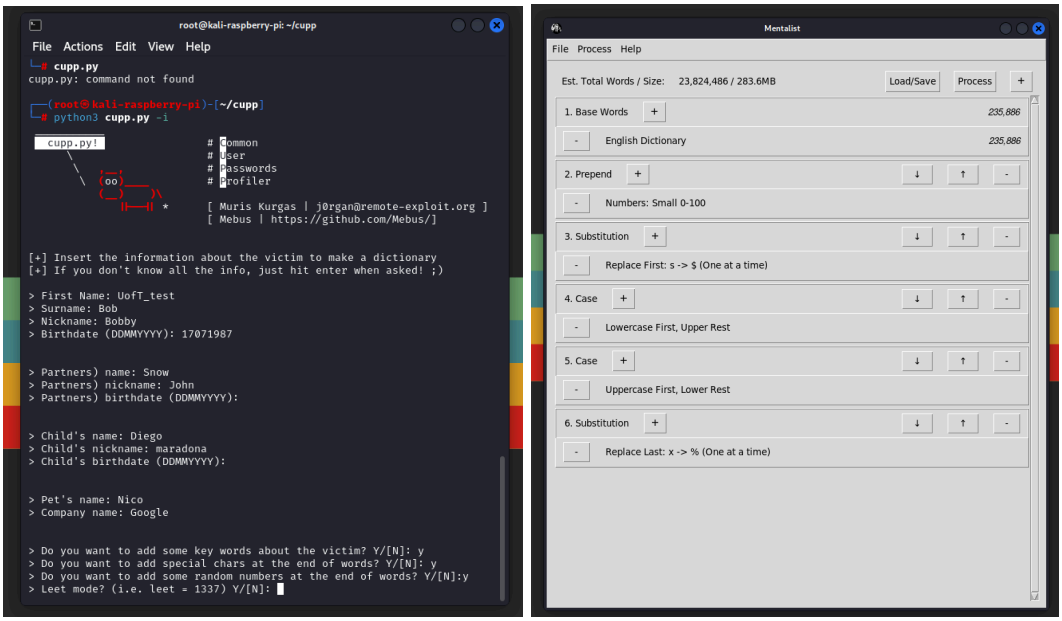
Aircrack-ng Dictionary attack on .cap file

Custom Wordlists

Custom wordlists are curated collections of words, phrases, or character combinations used in dictionary attacks and brute-force (when combined with hashcat for instance). They are tailored to specific targets or scenarios, enhancing the effectiveness of password-cracking efforts by focusing on relevant terms likely to be used by the target.

Social Engineering naturally plays a major role in this one. Especially if you use tools like CUPP combined with the Mentalist, where in order to generate passwords you need first to provide some information about the target, fairly easy to gather at the OSINT stage. CUPP needs installation, but there are some other programs that can also work: Crunch (Already pre-installed in Kali), is one of my favorites because it works by adding extra few advanced features, to include spaces and symbols in a particular order. Be aware of the size of these lists(Some of them exceed 80GB size). Tailoring these lists with specific rules is essential for efficiency in our tasks. One of the initial steps in pentesting is grasping the password policy of the targeted organization. Each task requires its own specific rules in wordlists. Cracking SSH and web app admin passwords differs greatly from decrypting Wireless WPA2 and WPA3 hashes (SHA-256). While dictionary attacks remain simple yet crucial, traditional brute-force methods are increasingly giving way to more sophisticated approaches like Mask Attacks.

CUPP + Mentalist



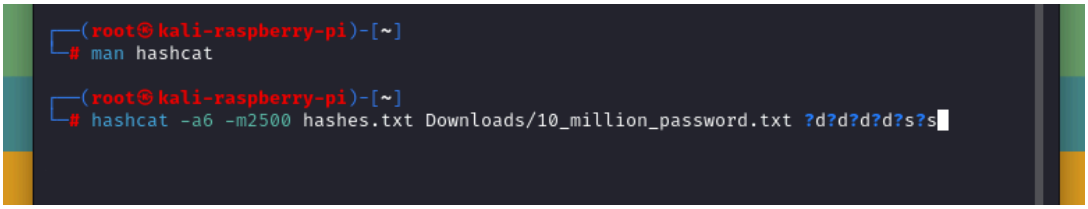
Left: Generating wordlist with Cupp.py interactive mode | Right: Enhancing the list with mentalist (23M passwords)

Hashcat (Wordlist + Mask)

Hashcat is renowned for its prowess in password recovery, supporting a wide array of hash types like MD5 and SHA-1. It features versatile attack modes including brute-force, dictionary-based, and rule-based approaches, catering to a range of cracking requirements. Notably, Hashcat leverages GPU acceleration to achieve impressive processing speeds, enhancing its effectiveness in decrypting passwords swiftly.

In a Mask attack, we aim to exploit human password patterns like combining a name with a year. By configuring the attack to focus on uppercase letters mainly in the first position(rarely in the second or third) we can reduce the keyspace to 237.6 billion combinations. At 100 million passwords per second, this would take only 40 minutes to crack.

Hascat: Hash types (2500 WPA/WPA2) and Attack mode (6 Hybrid Wordlist + Mask)



Hashcat mask: This one adds 3 digits and 2 symbols at the end of each password.

Mitigations

- Strengthen WiFi security by using longer and more complex passwords.
- Implement Wireless frame protection such as 802.11w to enhance security by validating deauthentication frames and rejecting spoofed ones.
- Ensure your router is updated to the latest WPS version.
- Activate WPS in "Locked" mode to prevent unauthorized access.
- Set up a threshold alert for Deauthentication (Deauth) packets to promptly detect and respond to potential attacks.
- Consider network segmentation, especially for enterprise environments, to isolate and mitigate potential breaches.
- Stay vigilant by keeping up to date with patches and addressing wireless vulnerabilities promptly.
- Conduct thorough research on routers before making a purchase to choose a model with robust security features.

Research References

- WPS Pin vulnerability published by Stefan Viehböck
- Hack the Box
- David Bombal Youtube Channel
- CVE 802.11 Vulnerabilities https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=802.11
- https://www.kali.org/tools/reaver/
- https://hackers-arise.net/
- https://www.kali.org/tools/wifite/
- https://www.opensourcedetective.com/posts/wifiphisher-rogue-access-point-attack
- https://security.packt.com/4-tools-to-create-your-own-custom-wordlist/