1   Exercises : Results & Analysis

1.1   Exercise 1 - Configure the access point through the laptop and the wireless adapter to monitor exchanged authentication data

T1: Setting up the Wireless Router

To being this lab, I set up the router we received and logged in through the browser using the default details set up on it already, as seen in Figure 1. Under wireless, we went ahead and started changing some of that default information, such as changing the SSID to "Hack3r" and the last two numbers of our JACard (see Figure 2, changing the wireless security to "WPA2 Personal", the WPA encryption algorithm to AES, the shared password to "password", and the Key renewal to the max, which is 99999 (these changes can be seen on Figure 3. We hit "Apply settings" and continued the lab with the updated router details.
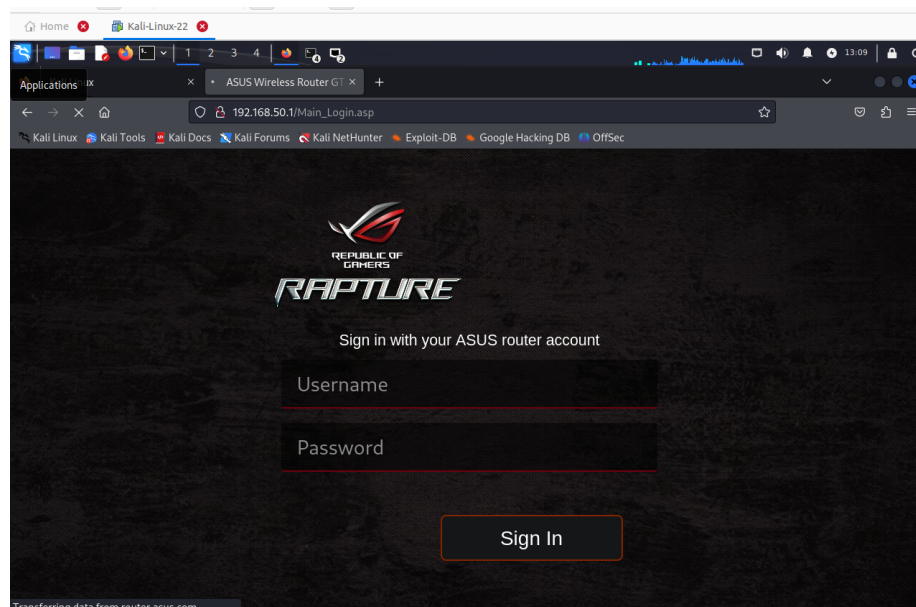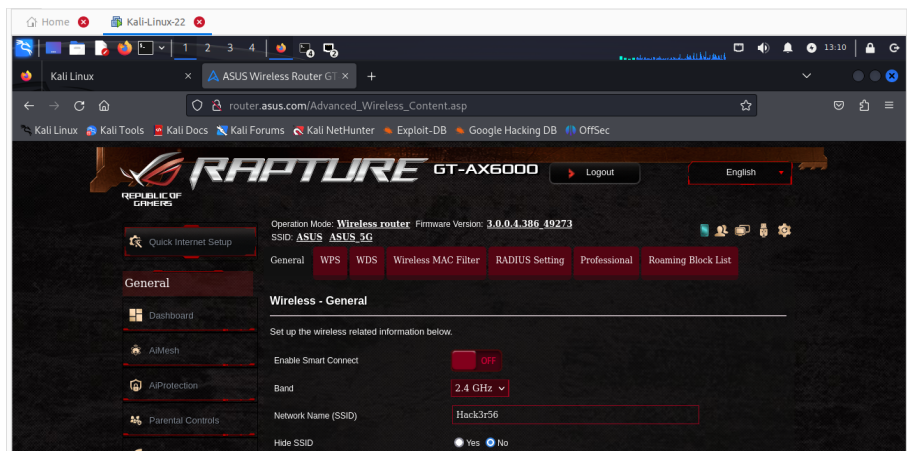

Figure 1: Login Page of router

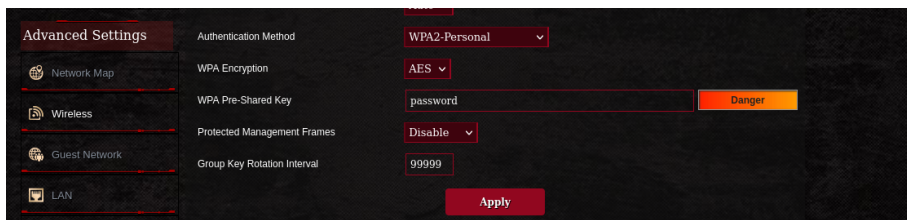Figure 2: SSID change to Hack3r56

Figure 3: Other default settings changed

T2: Setting up the Wireless Adapter

After starting up Kali Linux, we plugged in the wireless adapter we were provided and made sure it was connected to the VM and not the host computer (we will be using it in conjunction with the Kali Linux tools, hence why we need it on the VM). We checked the name of the adapter using ifconfig in the "wlan0" section (displayed in Figure 4).

The next step was to check whether our wireless interface was "managed mode" or not. We check this by using the iwconfig command, which quickly confirmed to us it was in "managed mode" (see Figure 5). Our goal then was to get it to read "monitor mode" instead, in order to capture the authentication packages and intercept the password of our router.

We typed in the command "sudo wifite" (and started the process). Figure 6 is what was displayed once we ran that command. From there, we found the SSID of our router in the list provided by wifite (see Figure 6) and noted the number by it which was "5". After selecting the number in the provided space, wifite executed a couple commands and set us to "Monitor mode", seen in Figure 7. Using iwconfig again came to show us we successfully changed over to "Monitor Mode" using the 2.467 GHz Frequency that matches the frequency set on our router in the first place (see Figure 8.

Figure 4: wlan0 seen on ifconfig

Figure 5: iwconfig to check for Managed mode

Figure 6: "Sudo wifite" command

Figure 7: Hacker56 - in NUM 5

Figure 8: iwconfig showing "Monitor Mode"

Q1: What is the difference between managed mode and monitor mode in wireless adapters?

A1: The different modes refer to what packets the wireless card in our computer picks up. According to Point [1], "Monitor mode" listens to all packets in our area (hence why we want it for this lab), while "managed mode" only listens to packets addressed to our MAC address.

Q2: What is WiFite and what is it used for?

A2: As you can see in figure 6 and according to Kali [2], it is a tool that displays encrypted wireless networks in our area and uses aircrack-ng, pyrit, reaver, and tshark to audit the them. The end goal of it is to identify and potentially exploit potential vulnerabilities in Wi-Fi networks, such as weak passwords or misconfigurations.

1.2 Exercise 2 - Cracking the WPA/WPA2 passphrase

T1: Capturing Four-Way Handshake (Authentication Packages)

In order to crack the password, we had to capture someone trying to gain access to the password of the router on Wireshark using out VM with the wireless adapter. We used a personal phone to connect it (see Figure 9) and captured this 4-way handshake using Wireshark. Typing in "eapol" into the filter narrowed down our search of our packets specifically, as seen in Figure 10.

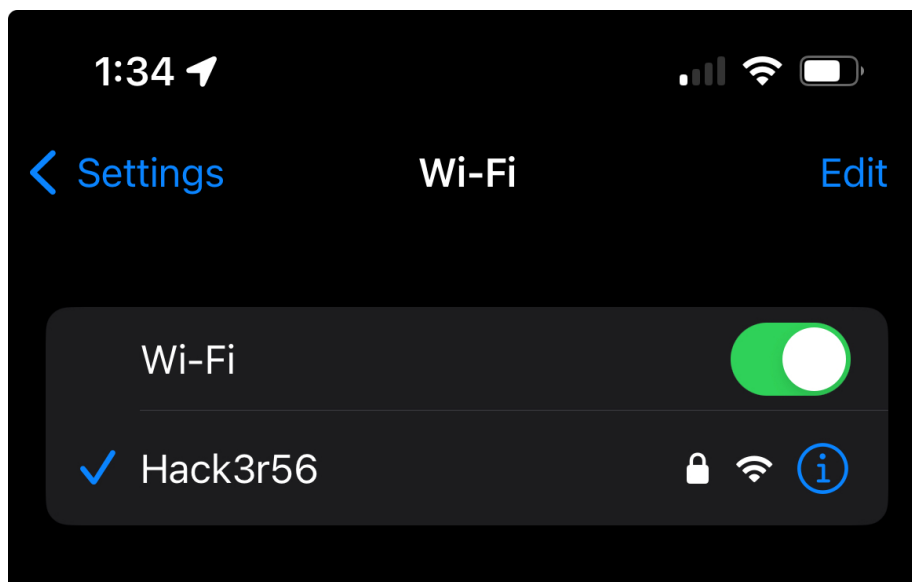The PCAP file can be found at Hack3r56.py.

Figure 9: Logging onto router using phone

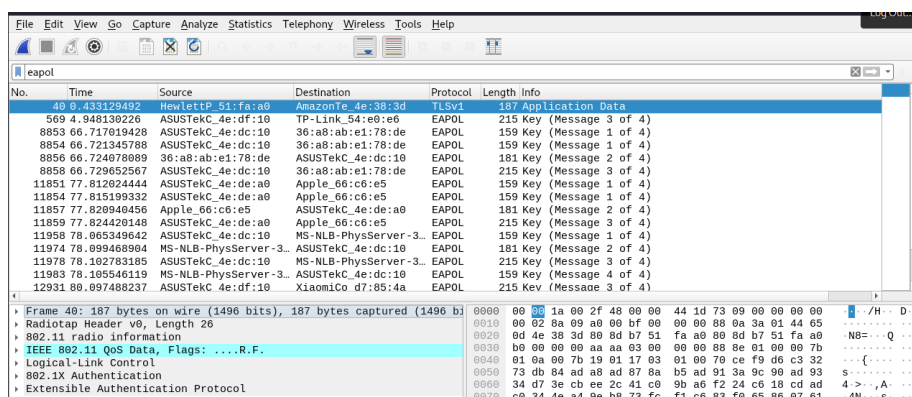Figure 10: "eapol" filter

T2: Cracking WPA2 WiFi Passphrase

We quickly learned to use aircrack-ng (see Figure 11) and ran the commands necessary to crack the weak password we set in the first place (see Figure 13).

```
┌──(kali⊛kali)-[~]
└─$ sudo aircrack-ng
[sudo] password for kali:

 Aircrack-ng 1.7  - (C) 2006-2022 Thomas d'Otreppe
 https://www.aircrack-ng.org

 usage: aircrack-ng [options] <input file(s)>

 Common options:

     -a <amode> : force attack mode (1/WEP, 2/WPA-PSK)
     -e <essid> : target selection: network identifier
     -b <bssid> : target selection: access point's MAC
     -p <nbcpu> : # of CPU to use  (default: all CPUs)
     -q         : enable quiet mode (no status output)
     -C <macs>  : merge the given APs to a virtual one
     -l <file>  : write key to file. Overwrites file.

 Static WEP cracking options:

     -c         : search alpha-numeric characters only
     -t         : search binary coded decimal chr only
     -h         : search the numeric key for Fritz!BOX
     -d <mask>  : use masking of the key (A1:XX:CF:YY)
     -m <maddr> : MAC address to filter usable packets
     -n <nbits> : WEP key length :  64/128/152/256/512
     -i <index> : WEP key index (1 to 4), default: any
     -f <fudge> : bruteforce fudge factor,  default: 2
     -k <korek> : disable one attack method  (1 to 17)
     -x or -x0  : disable bruteforce for last keybytes
     -x1        : last keybyte bruteforcing  (default)
     -x2        : enable last  2 keybytes bruteforcing
     -X         : disable  bruteforce   multithreading
     -y         : experimental  single bruteforce mode
     -K         : use only old KoreK attacks (pre-PTW)
     -s         : show the key in ASCII while cracking
     -M <num>   : specify maximum number of IVs to use
```

Figure 11: Manual of aircrack-ng

Figure 12: Passing parameters to aircrack-ng

Figure 13: "password" password is easily cracked

Q1: Explain the EAPOL packet type? When is it used? Why did we need to filter for it in monitor mode?

A1: According to Vocal [3], EAPOL stands for EAPoL Protocol – Extensible Authentication Protocol over LAN. It is the protocol that is used to login to access a network's resources.

Q2: What is Aircrack-ng tool? Other than cracking WPA passwords, what can it do?

A2: According to Aircrack-ng [4], "Aircrack-ng" is a tool has many different WiFi network security-centered functions. This includes cracking (like we used), monitoring, testing "WiFi cards and driver capabilities", and attacking using packet injection.

1.3 Exercise 3 - How would the attack work with a more decent password

T1: Repeating the Attack with a Stronger Password

To try to create a challenge, we changed the password to "eAPBXCIw71" and set out to crack the password. We repeated the steps in this lab to set the password and capture the 4-way handshake gained from connecting to the router.

The PCAP file can be found at Hack3r56BetterPass.py

Attempting to crack the password with the current wordlist yielded a fail, as displayed in Figure 14.

Figure 14: Stronger password is not cracked

T2: Finding a Better Wordlist

We downloaded a more comprehensive wordlist off of github to crack the stronger password.
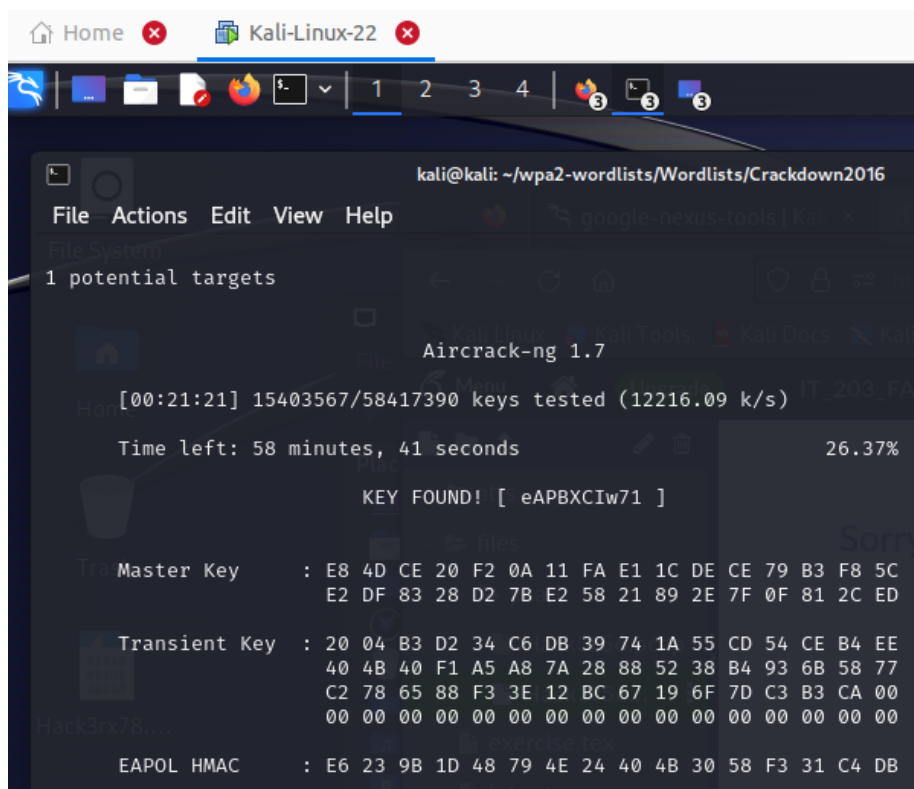
Figure 15: Stronger password is cracked

T3: Cracking the New Password

After running aircrack-ng with the better word list as a parameters for a few minutes, we were able to succeed in cracking it (see Figure 15).

Q1: Explain Each part of the screen below?

A1: Top presents the tool we are using "Aircrack-ng" and the version. Directly under, a password is being cracked and 48 keys out of 235 have been tested already (happening at 3832.89 keys per second). Under is the time left to try all keys and we have gone through 20.43% of the possible keys. The line under shows the current key being tested. In this case, the password "password" has been found. Under, the Master key is being shown in hex, as well as the transient key and the EAPOL HMAC.

Q2: Explain the changes we made to find the new password?

A2: To make the password stronger, we included lower and upper case

letters, as well as numbers. To find this stronger password, we used a more extensive wordlist we were provided with.

Q3: How does the new text file (/wpa2- wordlists/Wordlists/Crackdown2016/full.txt) help us?

A3: It is more extensive and comprehensive, holding exponentially many more possible keys than the common password wordlist.

Q4: What does hashing a password mean?

A4: According to auth0 [5], hashing a password is to using a hashing algorithm (such as SHA) to map data to a certain size. This ultimately "verifies the integrity of your password".

Q5: What are rainbow tables?

A5: According to Hill [6], rainbow tables are large, previously computed tables that contain the cache output of hash functions to decrypt hashed passwords into plain text.

Q6: What is HMAC? What does the HMAC value below represent?

A6: According to okta [7], HMAC stands for Hash-based message authentication code. This helps us achieve authentication by using a hash function and a key to make sure content remains private.

Q7: What does Aircrack-ng actually do to find the password?

A7: Aircrack-ng, is a tool that is used in deciphering WPA/WPA2-PSK passwords, as you can see in figure 11 it shows us various of commands that will be helped with the exploitation. To uncover the password, Aircrack-ng captures the authentication four-way handshake between a client and the wireless router. It then employs either a wordlist attack, systematically trying potential passwords from a provided list, or a brute-force attack, testing all possible combinations until success. Statistical analysis and algorithms assist in evaluating the success of each attempt.

1.4   Key Learning & Takeaways

From this lab, I was able to first setup the environment to test, including con-

figuring a router break into and Kali Linux to break into the router. Then, we were able to break an easy password with tools such as aircrack-ng and a Wireshark pcap file and then break into it again with a stronger password through the use of a wordlist and aircrack-ng.