# WPA Cracking with a Dictionary Attack

## Introduction

For my ethical hacking project, I decided to demonstrate how easy it can be to break into a network with an insecure password if that network is running WPA/WPA2. The main method of attack is through a dictionary attack, which entails trying a long list of passwords until the correct one is found. Listed here are the steps for accomplishing this task. First: We need to check around for nearby SSIDs. If you don't have a specific target in mind, this can be helpful to see possible targets in the area. Though, I would advise against attacking a real network in the wild because unauthorized cracking is illegal and could land you in prison or ordered to pay a fine.The second step goes along with the first step. If you know information about the target network, you can refine your search by putting it into the command.
As you'll see in the demonstration, I was able narrow my search using only the SSID and band as a starting point. Then I ran another search using the information I found in the first to narrow it down even further. Once you find the the target network, you must capture a 4-way handshake. This can be done by waiting for a device to connect, which can take hours or even days, or you can send a packet to force all clients to reauthenticate. Finally, We'll use cracking tools to compare the bits of the EAPOL packets (aka the handshake packets) to the wordlist. For this, I will be using Aircrack-ng and the rockyou.txt wordlist. This may seem like an obsolete attack because most people know to use a secure password, but you might be surprised at how effective this can be. According to NordPass, “123456” is the most commonly used password

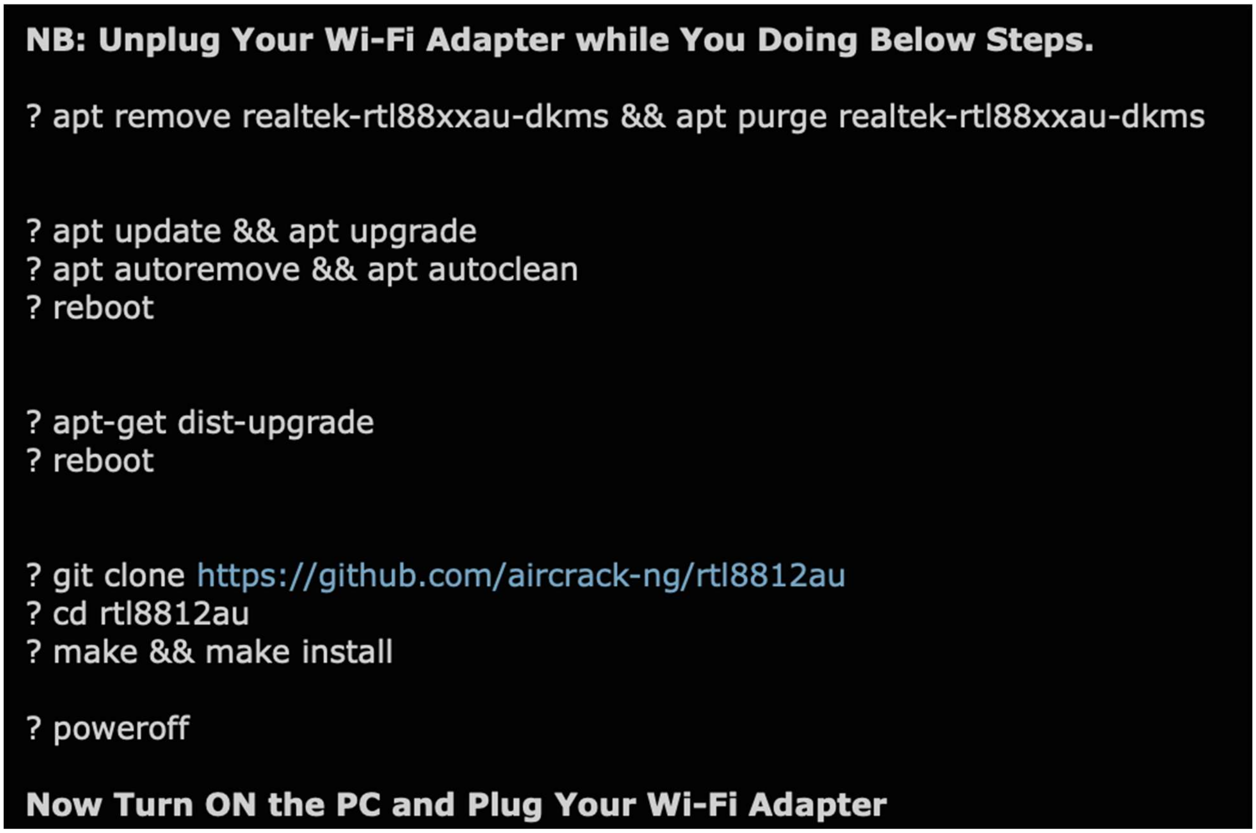
with over 100 million uses. That's a third of the entire US population. Clearly, password attacks aren't going away any faster than the use of insecure passwords.

## Lab Setup

To set up the lab, I used a 2021 MacBook Pro with 16 GB of RAM and an Apple Silicon M1 Pro. In case you're unfamiliar, this line of processors runs on ARM architecture, not x86-64 like the Intel based Macs do. This means I need to run the ARM64 version of Kali Linux through Parallels 17, a virtualization program that works in a similar way to VMWare. For the network interface, the onboard Wi-Fi card does not have the capabilities required for this type of attack, so I had to purchase a long-range adapter with monitor mode. The one I decided on was the Alfa Networks Dual Band AC1200. This adapter runs about $60 on Amazon, which may seem like a lot, but an adapter like this will would be a good long-term investment if you decide to do any sort of Wi-Fi analysis or cracking. This is one of the cheaper options for these kinds of cards, and it serves its purpose well. If you'd like to choose a different adapter, make sure the one you pick has monitor mode and packet injection capabilities. Dual band would also be useful so your adapter can handle both 2.4gHz and 5gHz Wi-Fi bands.

The Wi-Fi network I attacked was created by configuring a guest network on my home Router running under the 5gHz band. I called this network BRKM3P1S and asked my roommate to create a password without telling me what it is. I wanted as little information as possible to prove how easy it can be for attackers to break into a network. The only information I knew about the network prior to the crack is what's listed on this slide. All I truly needed to know, however, was the SSID.

Since I will not be using the host’s native OS for any of my network cracking, I configured the adapter to always connect to the VM in Parallels and never the host machine. In my case, I had to manually install the drivers using the commands shown here:



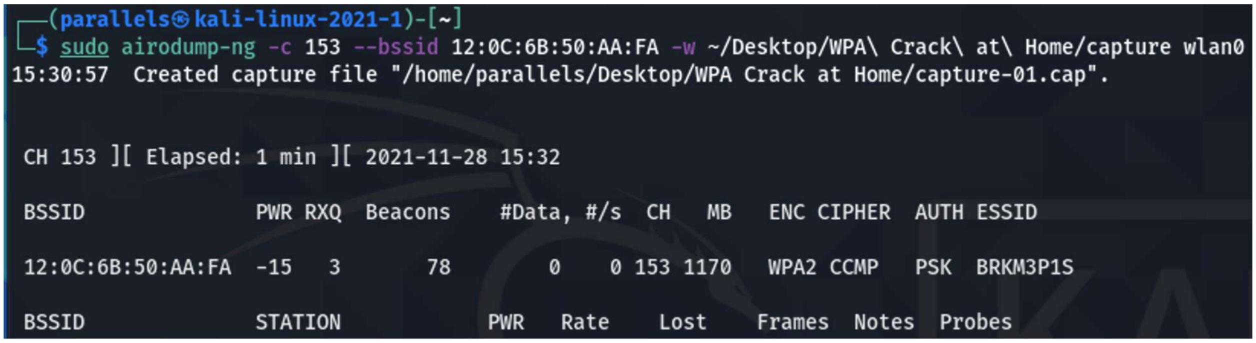
```
NB: Unplug Your Wi-Fi Adapter while You Doing Below Steps.

? apt remove realtek-rtl88xxau-dkms && apt purge realtek-rtl88xxau-dkms


? apt update && apt upgrade
? apt autoremove && apt autoclean
? reboot


? apt-get dist-upgrade
? reboot


? git clone https://github.com/aircrack-ng/rtl8812au
? cd rtl8812au
? make && make install

? poweroff

Now Turn ON the PC and Plug Your Wi-Fi Adapter
```

Depending on your lab setup, your installation process may be a little different, so conduct some of your own research to find a solution that works for you. Once you’ve set up the adapter, it would be a good idea to list the interfaces to verify the installation was successful.

To perform this attack, the adapter needs to be put into monitor mode using the airmon-ng command: airmon-ng start <Network Interface>. Monitor mode allows the adapter to scan for any Wi-Fi signals in the area. If not signed in as root, this will require sudo privileges. You may receive an error telling you that certain services could cause issues. Feel free to kill these processes prior to starting monitor mode if you wish.

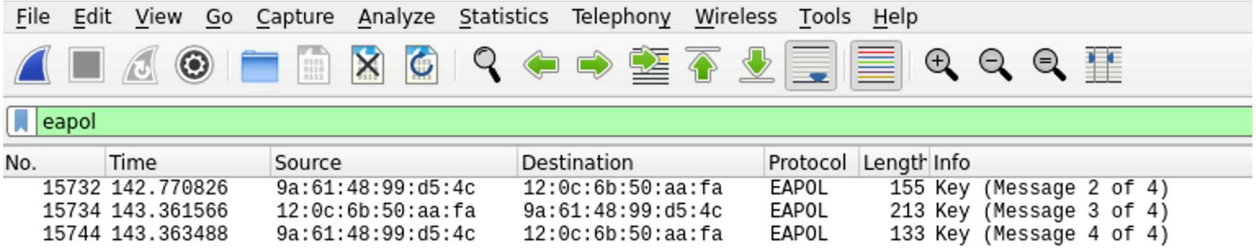
## Discovery

Now that the lab setup is complete, I can begin the discovery phase of the attack. Using the airodump-ng command, I input both the ESSID and band to target the network. Alternatively, you could just type “airodump-ng wlan0” and look at all of the SSIDs in the area. This will also tell you what channel the network’s running on, as well as the BSSID (MAC Address) of the router. These pieces of info will be useful later, so be sure to take note of these and store them somewhere you can access. Once you have this information, terminate this process. Since we now know the channel and the BSSID, we can run the command again with a more specific scope as well as output the capture to a folder I created on the Desktop. Leave this process running and we can move onto the next step, which is to capture the four-way handshake used for authentication.

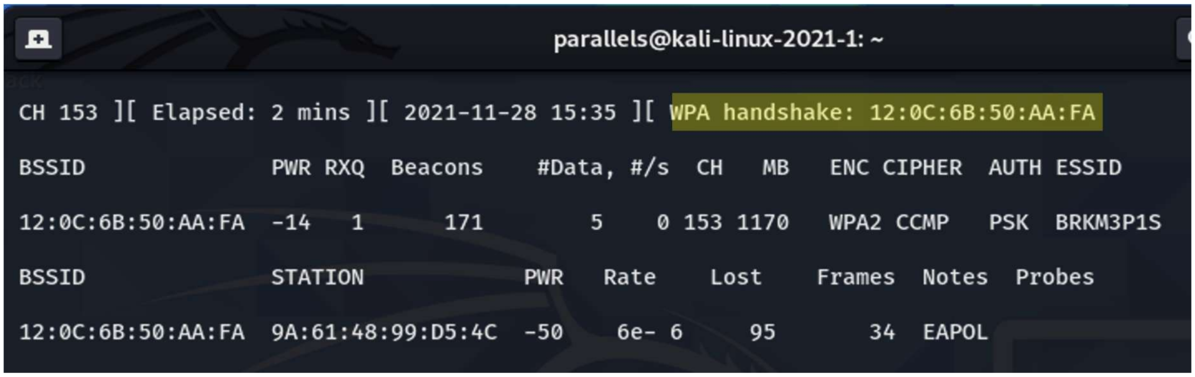


```
┌──(parallels㉿kali-linux-2021-1)-[~]
└─$ sudo airodump-ng -c 153 --bssid 12:0C:6B:50:AA:FA -w ~/Desktop/WPA\ Crack\ at\ Home/capture wlan0
15:30:57  Created capture file "/home/parallels/Desktop/WPA Crack at Home/capture-01.cap".


 CH 153 ][ Elapsed: 1 min ][ 2021-11-28 15:32

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 12:0C:6B:50:AA:FA  -15   3       78        0    0 153 1170   WPA2 CCMP   PSK  BRKM3P1S

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes
```

To capture the authentication handshake, we need to wait for device to connect to the network, then send a deauthentication packet to require the device or devices to reconnect. I asked my roommate to connect to the network to demonstrate this. As you can tell, the BSSID matches that of the router, and we now have the MAC address of his device. Looking at the capture in Wireshark shows the conversation with the router. For some reason, the first message was not captured, but it shouldn’t affect the results of this attack.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

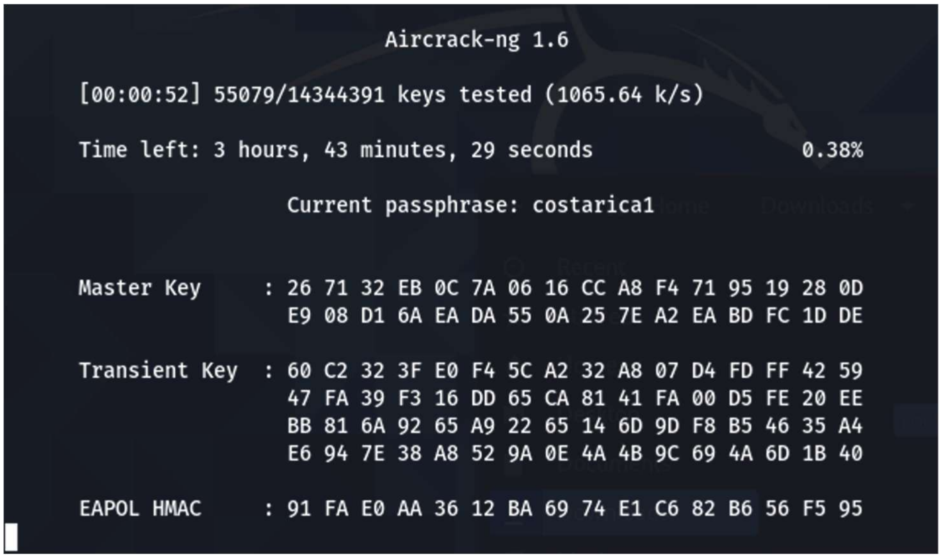
```
eapol
No.   Time        Source             Destination        Protocol  Length Info
15732 142.770826  9a:61:48:99:d5:4c  12:0c:6b:50:aa:fa  EAPOL     155 Key (Message 2 of 4)
15734 143.361566  12:0c:6b:50:aa:fa  9a:61:48:99:d5:4c  EAPOL     213 Key (Message 3 of 4)
15744 143.363488  9a:61:48:99:d5:4c  12:0c:6b:50:aa:fa  EAPOL     133 Key (Message 4 of 4)
```



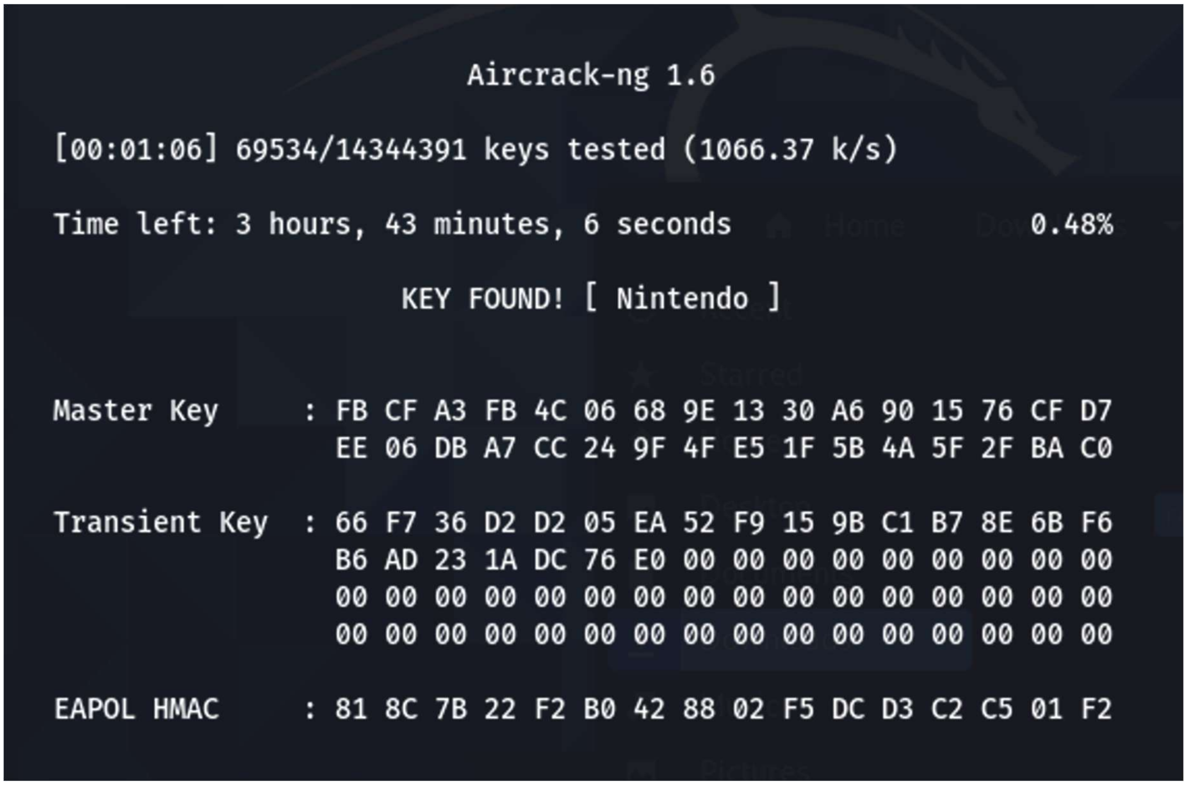
```
CH 153 ][ Elapsed: 2 mins ][ 2021-11-28 15:35 ][ WPA handshake: 12:0C:6B:50:AA:FA

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

12:0C:6B:50:AA:FA  -14   1      171        5    0 153 1170   WPA2 CCMP   PSK  BRKM3P1S

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

12:0C:6B:50:AA:FA  9A:61:48:99:D5:4C  -50    6e- 6    95       34  EAPOL
```

## Cracking the WPA Key

Upon running the command, it will run through the list and compare each line's bit values to the EAPOL messages we saw in Wireshark. If the program has to run through the entire list, it would take about 4 hours.



```
                           Aircrack-ng 1.6

      [00:00:52] 55079/14344391 keys tested (1065.64 k/s)

      Time left: 3 hours, 43 minutes, 29 seconds                0.38%

                     Current passphrase: costarica1


      Master Key     : 26 71 32 EB 0C 7A 06 16 CC A8 F4 71 95 19 28 0D
                       E9 08 D1 6A EA DA 55 0A 25 7E A2 EA BD FC 1D DE

      Transient Key  : 60 C2 32 3F E0 F4 5C A2 32 A8 07 D4 FD FF 42 59
                       47 FA 39 F3 16 DD 65 CA 81 41 FA 00 D5 FE 20 EE
                       BB 81 6A 92 65 A9 22 65 14 6D 9D F8 B5 46 35 A4
                       E6 94 7E 38 A8 52 9A 0E 4A 4B 9C 69 4A 6D 1B 40

      EAPOL HMAC     : 91 FA E0 AA 36 12 BA 69 74 E1 C6 82 B6 56 F5 95
```



```
                              Aircrack-ng 1.6

      [00:01:06] 69534/14344391 keys tested (1066.37 k/s)

      Time left: 3 hours, 43 minutes, 6 seconds                 0.48%

                           KEY FOUND! [ Nintendo ]


      Master Key     : FB CF A3 FB 4C 06 68 9E 13 30 A6 90 15 76 CF D7
                       EE 06 DB A7 CC 24 9F 4F E5 1F 5B 4A 5F 2F BA C0

      Transient Key  : 66 F7 36 D2 D2 05 EA 52 F9 15 9B C1 B7 8E 6B F6
                       B6 AD 23 1A DC 76 E0 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      EAPOL HMAC     : 81 8C 7B 22 F2 B0 42 88 02 F5 DC D3 C2 C5 01 F2
```

Voila! We’ve found the network key in just over 1 minute, even if my roommate decided to make it a little tricky by capitalizing it. It’s also interesting to see that the process only got through 0.48% of the list, which implies this is a commonly used password.

## Issues

Of course, I did run into some issues with this demonstration.

First, I had trouble finding the right driver for my wireless adapter, which was solved by multiple google searches. As mentioned earlier, the process missed one of the packets required for the 4-way handshake. This didn’t cause issues with the crack, but it is important to note Also, I found it difficult to find the rockyou.txt file in the Kali filesystem. I knew it was there somewhere but didn’t know where to look, so I downloaded the current list from the internet.

Finally, I did not properly set up the root account, so all of these commands required sudo to run properly. If you run all these commands as root, this shouldn't be a problem for you.

## Conclusion

For this demonstration, I am not going to exploit this network. I only wanted to show how an attacker can easily find the password to an insecure network. However, it's important to discuss how much further we could take this attack if we had malicious intention. Probably the most important target is that we could easily control IoT devices that are responsible for key aspects of the target's home. We could implement a ransomware attack that prevents the victim from making coffee, using AC or Heating, or even turning on lights without paying up. Similarly, we could access private files on network drives and then exfiltrate that data for financial gain. Streaming devices such as an Apple TV or a Chromecast allow us to display something on any connected screen in the house. Through better security practices, such as more secure passwords, along with improved security education, we can work to protect ours and our loved ones' assets. Since we live in an age where our whole lives are run through the internet, it's important to keep these practices in mind.