

Is Your Wi-Fi Really Protected?

Johnathan R. Santiago Laguna

Master in Computer Science

Jeffrey Duffany, Ph.D.

Electrical and Computer Engineering and Computer Science Department

Polytechnic University of Puerto Rico

Abstract — The purpose of this project will be testing a tool called ‘aircrack-ng suite’ and performed a penetration test to a private network. This suite is a collection of tools that allows you to assess the strength of your Wi-Fi security. This suite includes the tools airmon-ng, airodump-ng and aircrack-ng that are used to penetrate a home network cracking the key. For the project, I selected a small home network that only connect computers, smart tv, phones and printers. The owners of this network were aware of the attack and gave necessary permissions. This project intends to test how secure using WP2 protocol could be for a home network, create security awareness for people to protect their privacy and information.

Key Terms — cracking, cybersecurity, pen test, Wi-Fi.

INTRODUCTION

For this project the methodology I decided to use is the capture the 4-way handshake between the target host and the network router. What it’s a handshake? A handshake is when a device connects with an AP there’s where the 4-way handshake is executed. The handshake shares information between the AP and the device that is trying to connect to it. In Figure 1, we show the topology of the network select to attack in this project. The diagram shows how the devices are connected and the information of each connected host. The host Apple Computer is the victim who once I’m inside the network, uninvited, I’m going to sniff, monitored and capture the traffic that allow to be examined and possible extract credentials in plain text from it.

In wireless security exists 3 major security protocols that are used in variety of networks. The wireless security protocols are WEP, WPA and WPA2, serving the same purpose but being different at the same time. As explained by IPCisco [1], has

developed in 2006. It was advanced version of first WPA. Vulnerable parts of WPA become stronger with WPA2. WPA2 offered new encryption and authentication mechanisms to provide more secured networks. These mechanisms were AES (Advanced Encryption Standard) and CCMP (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol). These mechanisms were being used instead of previous mechanism TKIP. For interoperability, TKIP was also used but as a fallback. Dictionary Attacks are the most vulnerable part of WPA2 for passwords. For this purpose, the network we selected for this project is currently using WPA2. Theatrically speaking WPA2 is the hardest and tedious protocol to crack and access a network using it. Performing the dictionary attack to this network, in this project we can proof the point that WP2 has its vulnerability and is not secure for enterprise networks.

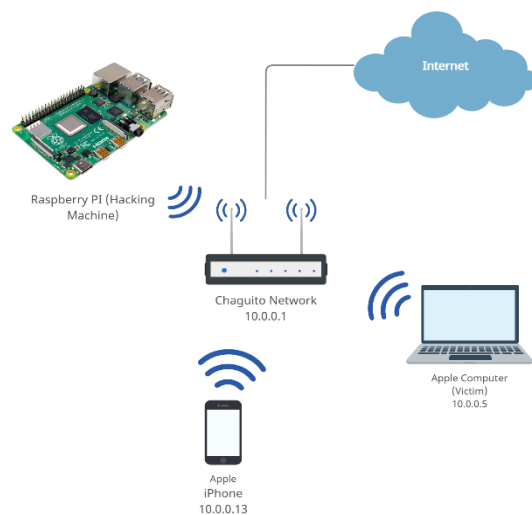


Figure 1
Target Network Diagram

REQUIRED TOOLS

1. Raspberry Pi running Kali OS.
2. USB Wireless Network Card
3. Wireshark
4. Nmap
5. Ettercap

PROCEDURE

The first step is that we began setting up our raspberry pi with a Kali Linux OS image. Kali linux is distribution aimed at advanced Penetration Testing and Security Auditing. Kali Linux contains several hundred tools targeted towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics and Reverse Engineering. Kali Linux is a multi-platform solution, accessible and freely available to information security professionals and hobbyists [2]. With an update OS we had just build our mini hacking computer, as part of the configuration process I validate that all tools mentioned are using the latest version. After we check all software requirements, we connect and configure our antenna capable of run-in monitor mode. Monitor mode, or RFMON (Radio Frequency Monitor) mode, allows a computer with a wireless network interface controller (WNIC) to monitor all traffic received on a wireless channel. Unlike promiscuous mode, which is also used for packet sniffing, monitor mode allows packets to be captured without having to associate with an access point or ad hoc network first. Monitor mode only applies to wireless networks. Monitor mode is one of the eight modes that 802.11 wireless cards can operate in: Master (acting as an access point), Managed (client, also known as station), Ad hoc, Repeater, Mesh, Wi-Fi Direct, TDLS and Monitor mode [3]. For this project the wireless usb used is an alfa wireless card model awus036acm. Most of the available usb card on the market are plug and play for the Linux environment and the company Alfa specializes in this type of hardware. We plugged in our usb wireless card. We must change our card mode to monitored as mentioned. To do this we run the

following command in a new terminal window “airmon-ng start wlan1” this changes the state of our card, and we can start the scanning process. We used airmon-ng, part of the aircrack-ng tool suite to change our interface mode. The Aircrack-ng is a complete suite of tools to assess WiFi network security [3]. It focuses on different areas of WiFi security:

- Monitoring: Packet capture and export of data to text files for further processing by third party tools
- Attacking: Replay attacks, deauthentication, fake access points and others via packet injection
- Testing: Checking WiFi cards and driver capabilities (capture and injection)
- Cracking: WEP and WPA PSK (WPA 1 and 2)

With our wireless interface in monitored mode will start to scan all available network on the area. This process just takes seconds to appear, after I see our target network, Chaguito Network, we stop the scanning pressing the ctrl + z letter combination. In Figure 2 it demonstrates the results of all detected networks.

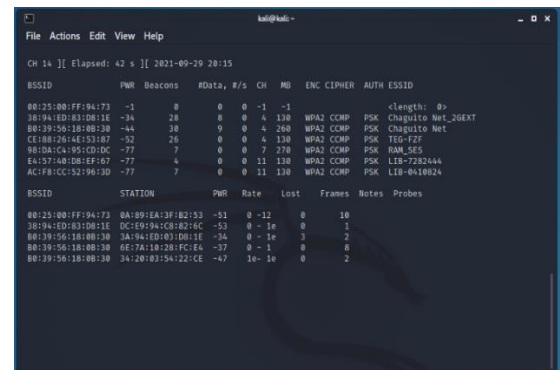


Figure 2
Scanned Networks

In order to monitor and intercept the traffic in the network we selected is important to gather the SSID of our network. SSID stands for “Service Set Identifier”. Under the IEEE 802.11 wireless networking standard, a “service set” refers to a collection of wireless networking devices with the same parameters. So, the SSID is the identifier (name) that tells you which service set (or network)

to join [4]. For our results this ID is on the first column from left to right. Using the SSID we can specify to our card using the tool aircrack-ng to only monitor traffic of that network. For our network the SSID was B0:39:56:18:0B:30. The algorithm of the tool start to monitor the network packets until we successful detect and capture the four-way handshake in the network traffic.

The process of detecting a four-way handshake on the network may take an indefinite time, a handshake occurs when exchanging 4 messages between an access point (authenticator) and the client device (supplicant) to generate some encryption keys which can be used to encrypt actual data sent over Wireless medium. In our scenario the four-way handshake toked approximately 10 minutes. To our luck the capture of the handshake was quite quickly due to a device was connecting to the network the moment we start monitoring the network. Now that we capture successfully our four-way handshake we can start the dictionary attack to crack the key to our target network. This type of attack is from the category of brute force attacks. In cryptography, a brute-force attack consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly. The attacker systematically checks all possible passwords and passphrases until the correct one is found. Alternatively, the attacker can attempt to guess the key which is typically created from the password using a key derivation function. Part of preparing a dictionary attack is to have in hand a file, called a dictionary, with all the possible combination of password. In technical terms is called a wordlist. The kali Linux distribution by default offers a file for this purpose called rockyou.txt. this This file consists of 14 million of possible passwords. The rockyou.txt by instance is a compressed file, in order to use it I had to uncompressed the file using the following command, “gzip -d rockyou.txt.gz”. this command uncompressed the file and save it in the same directory as a .txt file. As mentioned, the tool aircrack-ng will use the rockyou.txt wordlist trying each password one by one and is the password is in the list is going to show a message KEY FOUND.

An important step to successfully run the tool is that we save our capture file saved in the root directory. To verify that my 4-way handshake file was saved to the root directory with the name I specified I validate that we successfully saved the capture file running the “ls” command in the terminal. This command displays all the saved files in that directory. After running I see our file named cap1-02.cap in our root directory. Now I’m sure I have the right information and the tools sets; I was able to run the command the aircrack-ng tool to start the cracking process. In the command we specify the name of our captures file and the location of our wordlist file. The complete command was “aircrack-ng cap1-02.cap -w /usr/share/wordlist/rockyou.txt” the following command began to run without issues. There’s no exact time to successfully decode the network key, the cracking process could take up to 2 to 14 hours to crack it. For this project I let run and the process toked 3 hours 49min, but the cracking process was successful. In Figure 3 I demonstrate how aircrack-ng tries every passphrase.

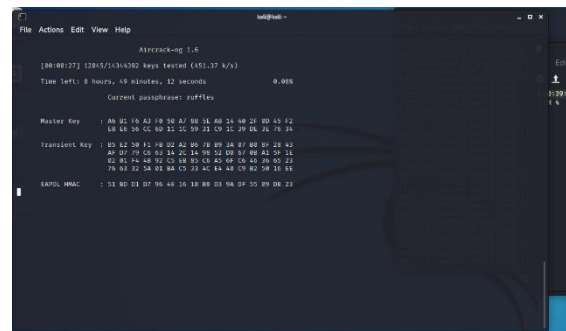
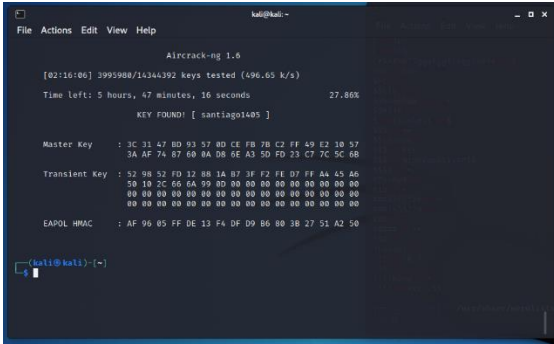


Figure 3
Password Cracking

The key was a simple text password. The password for the network was “santiago1405”. Figure 4 shows when the password was found and display the message KEY FOUND. It just used 27% of the wordlist at successfully crack the password.

Now that we have the key, we can connect to the network access point to start detecting, planning and executing other types of attack. The first step is to detect and determine the network topology as we show in Figure 1. The main device we need to identify is the router and its address. Knowing this gave us the whole configuration of the network

including the subnet configured. To determine the router address I execute a simple command in the terminal, “ifconfig” this command show the network information including IP, subnet mask and gateway.



```
kali@kali:~$ aircrack-ng 1.6
[02:10:00] 3995988/1244392 keys tested (496.65 k/s)
Time left: 5 hours, 47 minutes, 16 seconds      27.86%
KEY FOUND! [ santiago1405 ]

Master Key   : 3C 31 47 8D 93 57 8D CE FB 7B C2 FF 49 E2 3D 57
              3A AF 74 87 68 8A 08 0E A3 5D FD 23 C7 7C 5C 6B
Transient Key: 52 98 52 FD 32 88 1A B7 3F F2 FE 07 FF A4 45 A6
              5B 1B 2C 66 6A 59 8D 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   : AF 96 05 FF DE 13 F4 DF D9 B6 80 38 27 51 A2 50

kali@kali:~$
```

Figure 4
Key Found

For our network the router address was 10.0.0.1. as I mentioned knowing the router address, gave us the power to scan the whole network, in that scanning process we can detect all connected devices and capture the traffic of a host, in this case our next victim. For the next attack and for penetration purpose I executed a man in the middle attack to captures all packets from the host to the router. In cryptography and computer security, a man-in-the-middle, monster-in-the-middle, machine-in-the-middle, monkey-in-the-middle, meddler-in-the-middle (MITM) or person-in-the-middle (PITM) attack is a cyberattack where the attacker secretly relays and possibly alters the communications between two parties who believe that they are directly communicating with each other, as the attacker has inserted themselves between the two parties [4]. For the purpose of scanning the network and detect all connected hosts I used the tool nmap to scan the network and retrieve information that help me identify the device like IP address, MAC address and hostnames.

Nmap (Network Mapper) is a free and open-source network scanner created by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich) [5]. The tool is used to discover hosts and services on a computer network by sending packets and analyzing the responses. To the start the scan I had to run the command “sudo nmap 10.0.0.1” this

command will scan all the connected devices to the router. After the scan show the results and information, for the purpose of this project I selected a connected host, an apple computer with the IP address 10.0.0.15 to execute the man in the middle attack. The type of man in the middle attack I performed on this project was ARP Poisoning. This type of cyber-attack is carried out over a Local Area Network (LAN), our target network, that involves sending malicious ARP packets to a default gateway on a LAN in order to change the pairings in its IP to MAC address table. ARP Protocol translates IP addresses into MAC addresses. In simple words, I redirect all the traffic from my target device, the apple computer, to the internet to pass through my machine, the raspberry pi, first instead of the network router. This rerouting gives me the opportunity to sniff and be able to capture all the packets containing information that I can analyze. For the purpose of this attack the analysis intentions is to retrieve any credentials from the packets. If credentials are extract from the packet, we successfully compromised the network, and accounts of the victim.

For the ARP poisoning attack, the tool I used was Ettercap. Ettercap is a comprehensive suite for man in the middle attacks. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols and includes many features for network and host analysis. With the running ARP poisoning, I start to see all the traffic from my target host. After a few minutes of capturing all the traffic we stop the process and saved it on my root directory. Figure 5 shows all the traffic is reading from the apple computer, this means the man in the middle attack was working.

The saved file that was generated I called it cap1-02.cap. now, being a .cap extension file it can be open with a packet analyzer. For this project I used the most notorious and tested tool: Wireshark. Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education.



Figure 5
Traffic from Computer

Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues. It lets you see what's happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions. Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998[6]. To begin to analyze our file, we must run the command 'sudo wireshark cap1-cap02.cap' this command will open the Wireshark application with administrative privileged. There is an innumerable way to modify or customize our packet analyzing process. For this project, in order to make the analyzing process an easier one I applied filter in the command bar to see only traffic from our target host, the apple computer, to the outside network, this filter opt out any internal traffic e.g., from the host to a network printer or private server. When all traffic was filtered, we can visualize with who and how the host interact with external connections and the protocol used. In the Wireshark version 3.1 or older is available a new feature that detect credentials in the packets. In theory for the scope of this project this feature is handy. The option was available in the menu bar under the tools menu called 'credentials'. For the packets we capture from our victim it was not possible to retrieve any credential data. In Figure 6, I show a screen capture that Wireshark was not able to capture credentials form the captured packets.

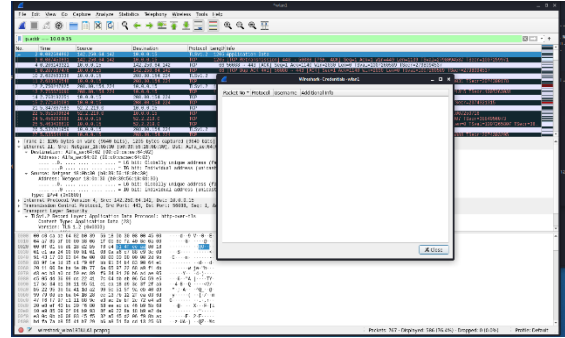


Figure 6
No Password Detected

CONCLUSION

The results of this project were impressive from a network security perspective. We successfully crack the password of a home private network, manipulate the traffic for one of the connected devices and almost gather accounts credentials from the user. The tools I tested I can validate that are powerful, robust, and accurate. I think this project can demonstrate that any security is not perfect security nor can be trusted.

From the non-technical user standpoint, a project like this can opened their mind to be more serious and aware about things they can do to reinforce their security. The digitalization is transforming the world in light speed, network security and other aspect of computer security are in the highest demand being the number one treat to any business, person or entity. Security professionals are making the best effort to create awareness in the people in how important security is and what are the best practices to implement in our lifestyle. But most importantly is our responsibility to keep learning and give security the importance in should have.

As a lesson learned from this project, we see the importance of creating complex password for any digital account or device. A complex password must be created containing combination of letter, numbers, symbols and a length of more than 10 characters. This project fulfilled the purpose of demonstrating that WP2, being the most secure of all network protocols, can be vulnerable and cracked. Most importantly, the results of this project serve as inspiration and open new opportunity in research and

investigation in newer protocols being in develop and that will be the new standard in a near future. We are living in a world that just one single vulnerability is all an attacker needs.

REFERENCES

- [1] Ipcisco. "Wireless security protocols." *Ipcisco.com*. Sept. 15, 2020. [Online]. Available: <https://ipcisco.com/lesson/wireless-security-protocols/>
- [2] S. Klimaszewski. "Raspberry pi." *Kali*. Oct. 14, 2021. [Online]. Available: <https://www.kali.org/docs/arm/raspberry-pi/>
- [3] Aircrack-ng. "Aircrack-ng." *Aircrack-ng.org*. Sept. 18, 2019. [Online]. Available: <https://www.aircrack-ng.org/doku.php?id=aircrack-ng>
- [4] Hoffman, C. "What Is an SSID, or Service Set Identifier?" *How-To Geek*. Dec. 5, 2017. [Online]. Available: <https://www.howtogeek.com/334935/what-is-an-ssid-or-service-set-identifier/>
- [5] Nmap.org. "Nmap network scanning." *Nmap.org*. 2000. [Online]. Available: <https://nmap.org/book/man-host-discovery.html>
- [6] Wireshark. "Go Deep." *Wireshark*. 2020. [Online]. Available: <https://www.wireshark.org/index.html#about>
WS