Senior Project

Edward Ayala| Report 1

Name idea: WARP Tool (**W**ireless **A**ll-In-One **R**econnaissance & **P**enetration)

This project will be a tool that takes advantage of vulnerabilities of wireless networks set up by companies such as Charter Spectrum (potentially Xfinity, and AT&T). The main vulnerability this tool will be exploiting is the structure of the default passwords that these companies configure when installing the routers in customer's homes. This tool is also taking advantage of the fact that most consumers are not tech savvy and will not go out of their way to change the default password, mostly because the passwords are easy to remember phrases and they are printed on the routers. Based on my observations, Charter Spectrum's default password structure is an Adjective-Noun-Three digits, while Xfinity's is a similar combination, and AT&T's is the most complex with 12 alphanumeric characters including symbols; making it the hardest password to crack with the current methods available. Complex passwords like AT&T's call for different approaches for auditing the network.

The tool will be a Linux command-line tool that incorporates other tools that will help with the process of grabbing the wireless credentials. The main outside tools that I will be using are part of a suite of Wi-Fi security auditing tools called Aircrack-ng. The tools I will be using include Airmon-ng, Aireplay-ng, and Aircrack-ng. Each of which play a part in the process of auditing wireless networks. Another potential tool that I will attempt to implement in the project is called Hashcat, which works very similarly to Aircrack-ng but it takes advantage of other computer hardware allowing for faster password cracking.

The project will consist of writing code that either creates a wordlist using predefined words or fetches a wordlist from online sources. The program will then combine the words into the various password structures (Adjective-Noun-###) that will be used within the other tools for cracking the captured passwords. While the program compiles the wordlist combinations on-the-fly, it will also attempt to crack the captured wireless password. This is to save memory and storage space. Before the main wordlist process is initiated the program must use tools from the Aircrack-ng suite to first find suitable networks to audit, de-authenticate the connected devices of the network, capture the encrypted password as the devices reconnect to the network (known as the handshake), then the wordlist comes into play as it is used to crack the passwords.

This project's goal is to shine a light on the dangers of default passwords as well as the disadvantages of not having the basic knowledge of managing a home network for home security. Due to the fact that if a malicious agent was able to break into a home network, they would be able to access smart devices such as cameras, televisions, and door locks. My hope is that this project will teach those who are not aware of these dangers to secure their own networks and be wary of the other potential dangers in wireless security.

- Aircrack-ng Wiki:
  - https://github.com/aircrack-ng/aircrack-ng
- Hashcat
  - https://github.com/hashcat/hashcat