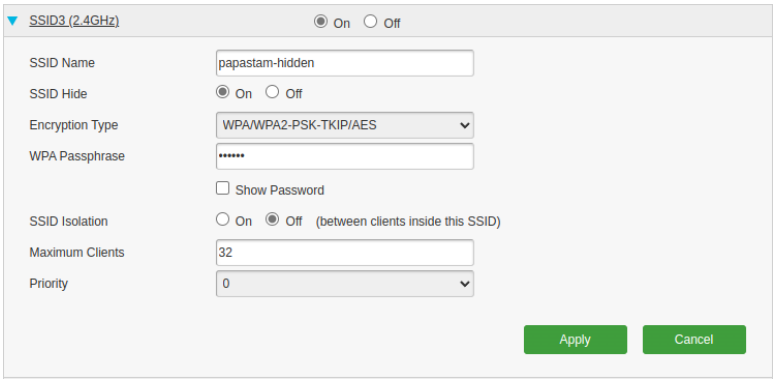
# HY455 Cyber Security
# Assignment 1
# Chris Papastamos (csd4569)

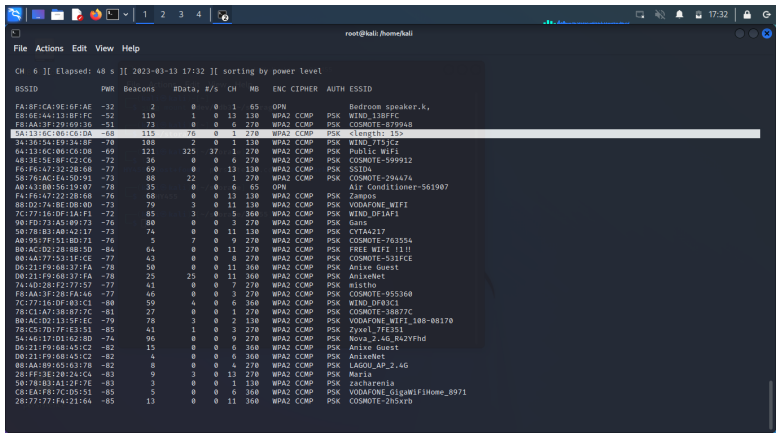
## 1. Pawning SSIDs

For this task I created a hidden SSID from my home router that I could later pawn using kali Linux:



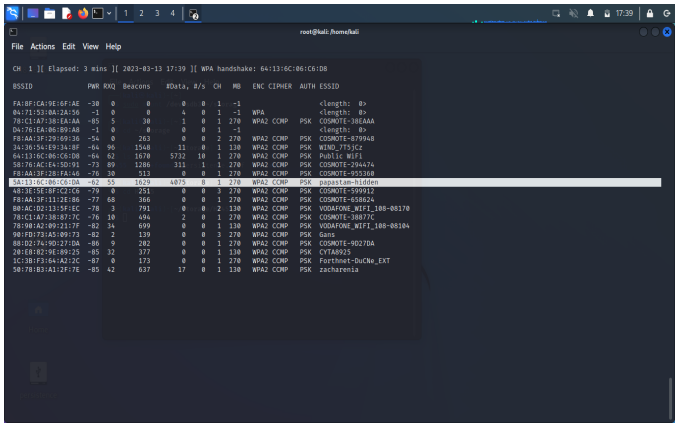

In my laptop first of all I turned my wifi antenna into monitoring mode using the following three commands: *$ sudo ifconfig wlan0 down; sudo iwconfig wlan0 mode monitor; sudo ifconfig wlan0 up*

After that, using airodump-ng (*$ airodump-ng wlan0*) I captured the nearby networks, including one hidden with a SSID of 15 characters (This should the SSID I created).

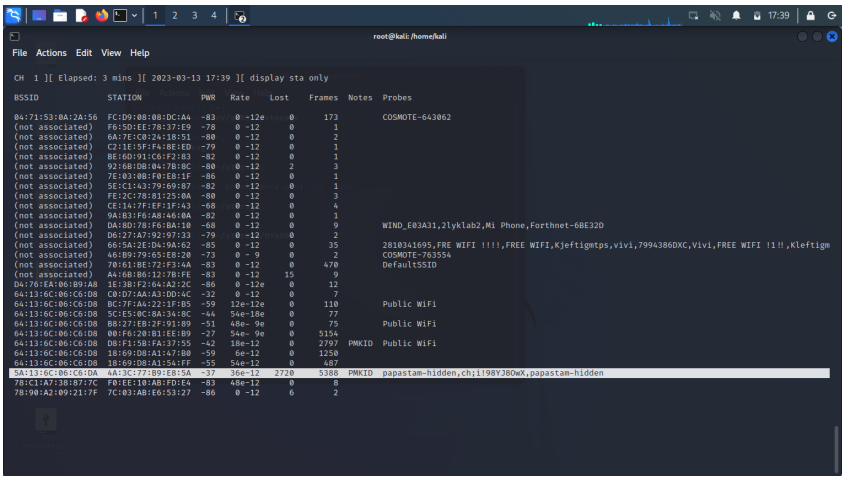


Because there are a lot of results in airodump we will restart it showing only channel #1 SSIDs (*$ airodump-ng –channel 1*).For demonstration I will now connect my mobile phone to the hidden WiFi network so that airodump can get the name and display it (this could also be done by de authenticating an already connected station so that it tries to reconnect)

Here we can see the name of the hidden wifi:



And here my mobile phone exchanging packets with the station:

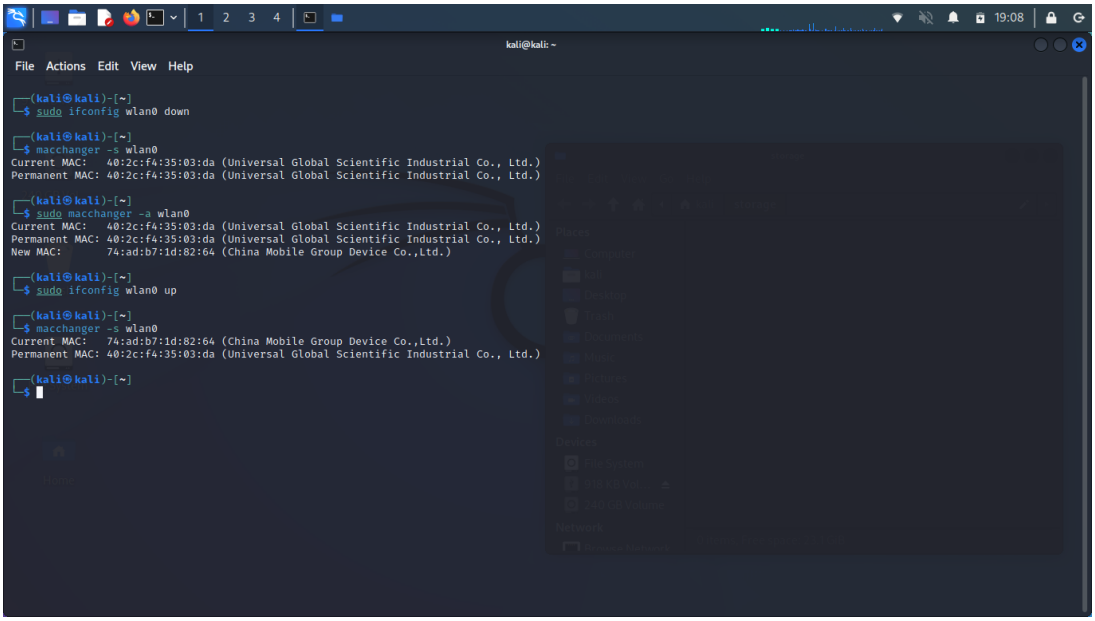



## 2. MAC spoofing

A MAC address is a static address for each interface of a device and for that reason the major way of recognizing a network device is through its MAC address. Therefore, a device which can change it's MAC address can pretend to be another device.

MAC spoofing is the process of changing the MAC address of an interface. This is done for multiple reasons including impersonating other devices on a network or even avoiding filters of a router which clock or restrict access to a resource or the Internet. A commonly used tool in Kali Linux is *macchanger* which allows a user to change an interface's MAC address

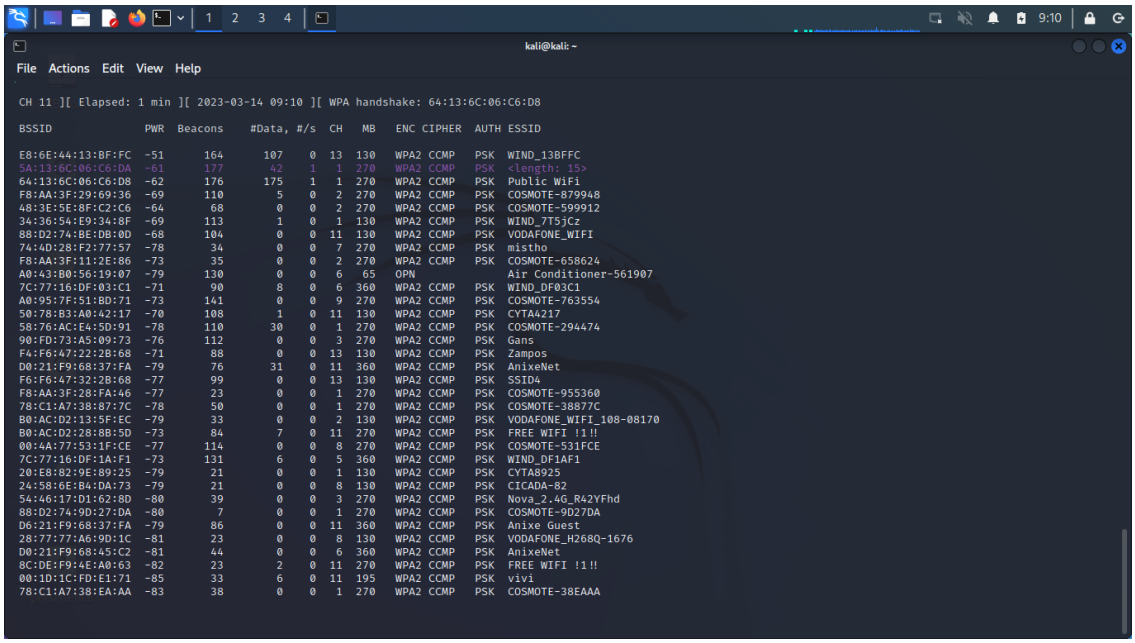
For demonstration I will start by taking down the interface that I want to change its MAC. After I user *macchanger -a wlan0* to change the MAC address and then take the interface up once again.



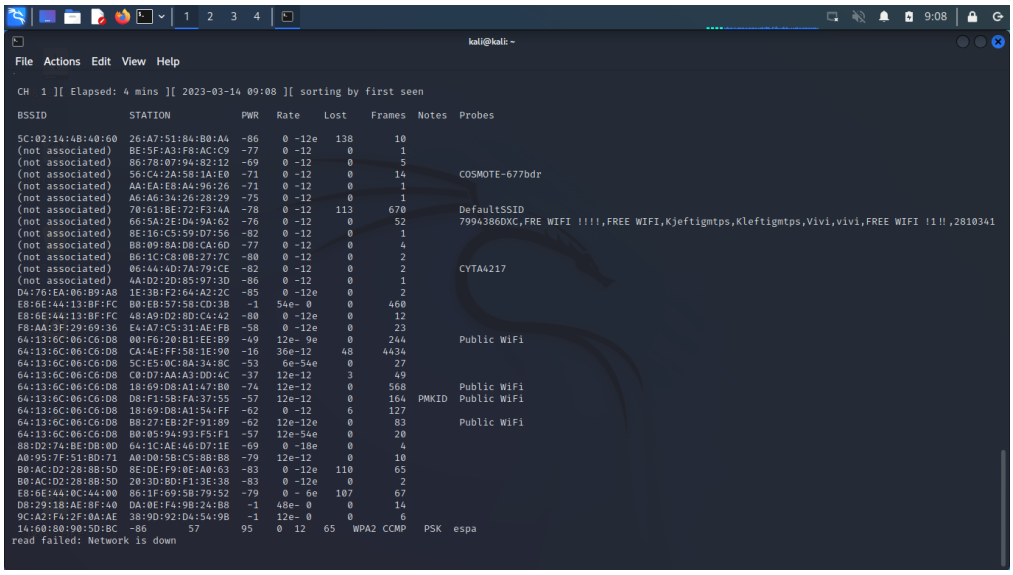
## 3. Scanning for wireless networks

The most commonly used command to monitor WiFi Access Points and Stations is airodump-ng. Using this tool you can gather useful information like hidden networks (as demonstrated in the first exercise), the proximity of a AP, stations and the APs that they are associated, stations “looking” for their previously connected APs. For airodump-ng to work you need a WiFi antenna in monitor mode

In the following screenshot we can see airodump-ng output for the APs only near my house sorted based on their proximity to my laptop. In this view we can also acquire the AP’s MAC (BSSID), the times the AP has sent a beacon (Beacons), the proximity (PWR), the encryption (ENC), the authentication used (AUTH) and obviously the SSID (ESSID) (You can also see my hidden SSID from the first exercise):




Cycling through displays using the ‘a’ key we can see the stations reaching for beacons. Some stations are not associated with any AP , which shows in the BSSID column as “*(not associated)*”. Sent and lost frames can be acquired from their respective columns, as well as the proximity of the station from the PWR column but most importantly the station’s MAC address from the STATION column.



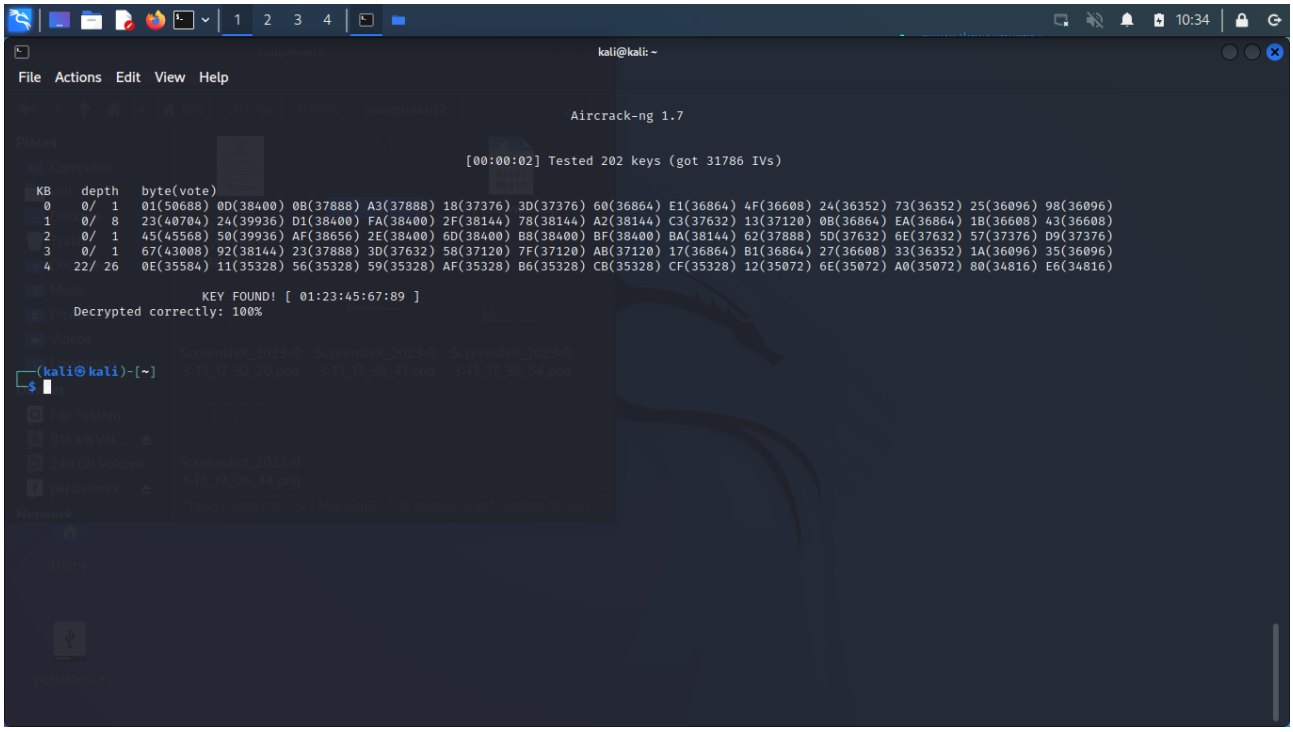
An interesting entry is also captured in the screenshot (the one with multiple Probes while not associated with any AP). This station is not connected to any AP and is looking for any previously connected AP by probing its known APs. This is a very interesting find because we now can create an AP with any of the probed SSID and the station will connect automatically thus giving us the possibility of a man in the middle attack

## 4. Attacking WEP

The WEP protocol is a very old protocol (since 1997) and according to a vulnerability which surfaced around 2001 the protocol's security wasn't that hard to crack. Basically the algorithm uses a 24-bit Initialization Vector and a 40-bit key to form a 64-bit key. According to a 2001 study the cryptanalysis of the WEP key and the use of the IV is fairly easy to exploit. Some countermeasures have been recommended including different WEP keys for different users.

In order to capture the traffic of any Access Point (in our case, **HY455-wep**), we need to run airodump-ng on a wireless interface in monitor mode and store it in a .cap file. This can be done using the command *$ airodump-ng wlan0*. After we get the AP's BSSID (MAC address) we can run *$ airodump-ng -b <AP's BSSID> -w <file prefix> wlan0* only for this AP in order to collect information about it, write it to a file and later crack it using aircrack-ng.

The usage of aircrack-ng for AP's using WEP is very simple. You just need to run the tool giving as an argument the .pcap file from the capture described before. Aircrack-ng will run for a short period and then will come back with the result of the WEP KEY as in the screenshot below.



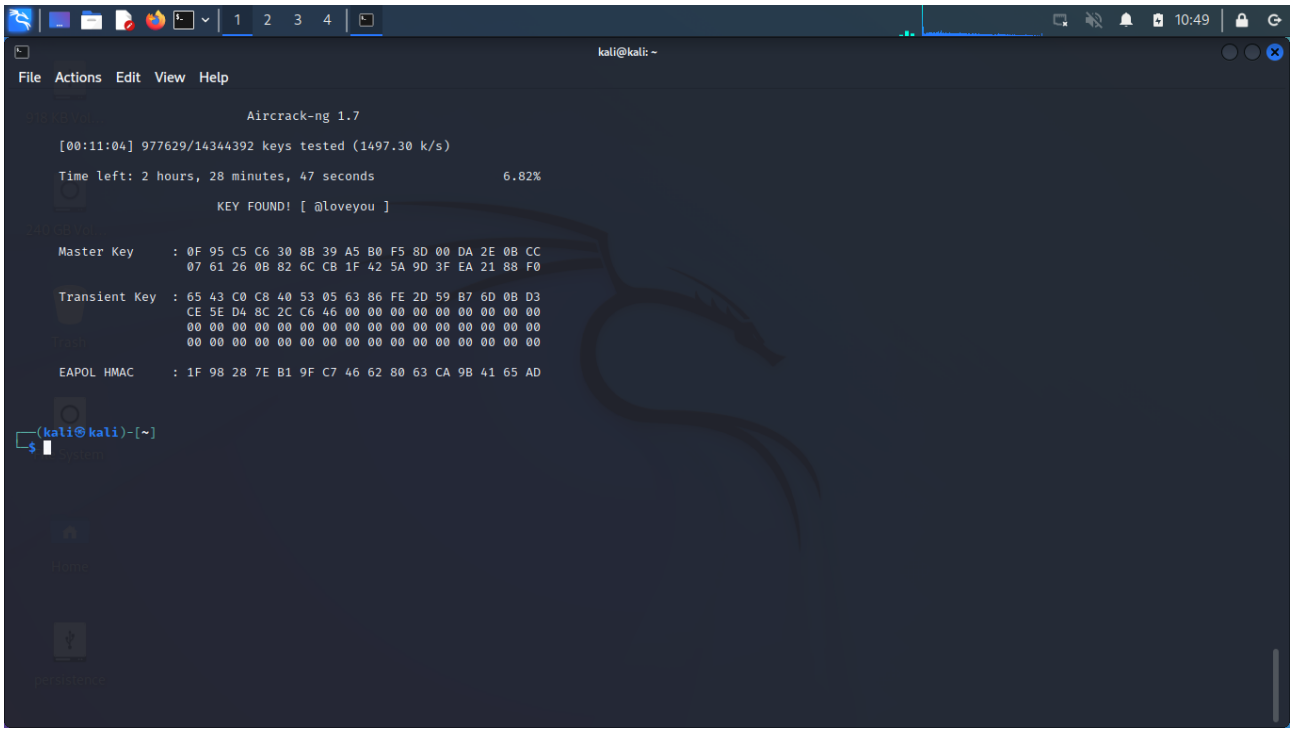

## 5. Attacking WPA

Same as the WEP protocol, to crack a WPA protocol we first need to listen to the conversation between an already connected user and the AP.

For this attack the attacker needs to capture a 4-way handshake of the AP and the device. This can either be captured while sniffing the traffic and someone randomly connects to the WiFi, or it can be forced by deauthenticating an already connected device. To deauthenticate a device the attacker sends a couple of disassociation packets to the AP letting it know that the device needs to be disconnected. The device notices that and tries to connect again by a 4-way handshake that the attacker captures. The keys used for this handshake can then be brute-forced with a list of commonly used wifi passwords until the WiFi password be found.

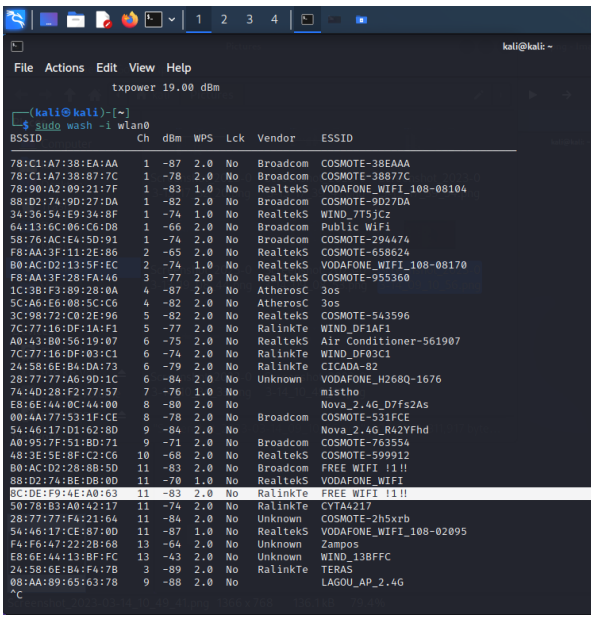
For these steps we are going to use airodump-ng to sniff the traffic and write it to a file like in the previous exercise, and then pass that file to aircrack-ng as well with the rockyou.txt which is a huge directory of commonly used passwords. Let that cook for a little bit and if a key matches, the WiFi key is found

In this screenshot we can witness aircrack-ng crack the password of HY455-wpa and return the password “@loveyou”




## 6. Attacking WPS

For the WPS attack to work a router must have WPS enabled so we can brute-force the WPS pin. The commands needed for this attack are wash and reaver. Wash is a tool that searches the available APs and displays all the WPS enabled APs. It also shows what version of WPS the AP has enabled and if the WPS is locked. When you pick a target from the wash output you can then run a Pixie Dust attack which basically brute-forces the 8 digit pin of the AP (The last digit is for error correction so if calculated that leaves us with 7 digits (10^7 possibilities)).I personally tried this attack but reaver seems to be stuck on trying the first pin (12345670) and retrying it till infinity.




The most common countermeasure for this attack is for the defendant to turn the WPS completely off as it is not commonly used on everyday households and as a result it is just a threat to the WiFi owner (If not used).