INFSCI 1600: Security and Privacy

Fall Semester 2021

Project 3 Part 2- Wifi Hacking

November 17, 2021

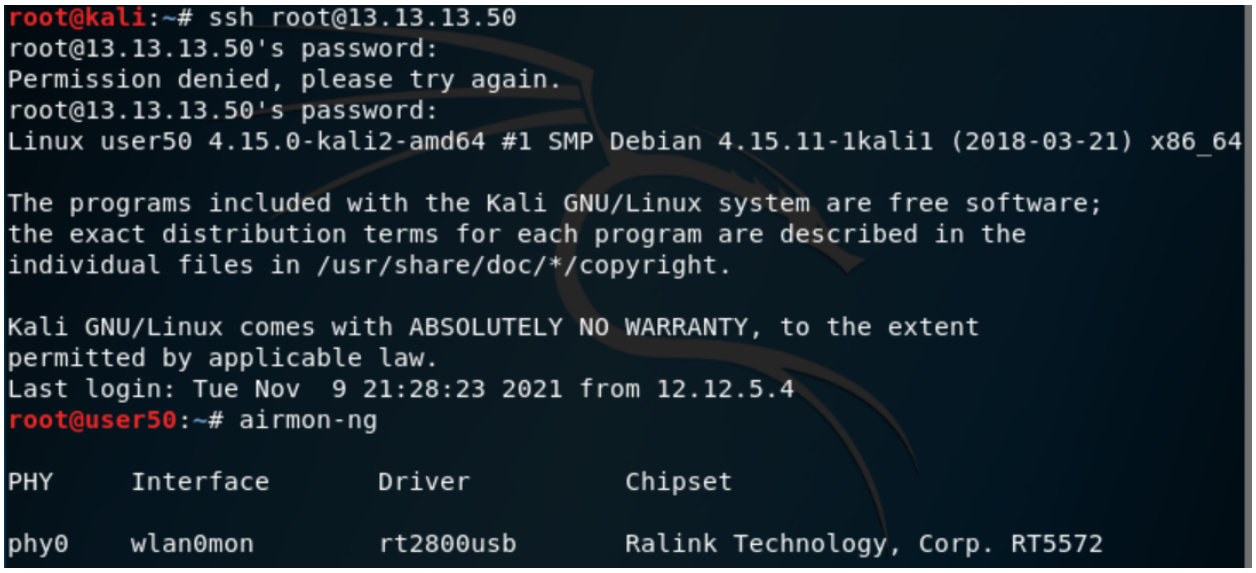
Jonathan Azcona

Section 1: Report on exploiting Rome AP

1.1 The bssid for Rome is **_B0:BE:76:08:BE:0C__** .
1.2 The channel for Rome is **__9___** .
1.3 The manufacturer of Rome is **_Tp-link technologies_** .
1.4 The HEX key (a.k.a. password) for Rome is **_12:34:56:78:90__** .
1.5 This attack took me/us **_5_** hours to perform.
1.6 Step-by-step documentation of how you performed the exploitation

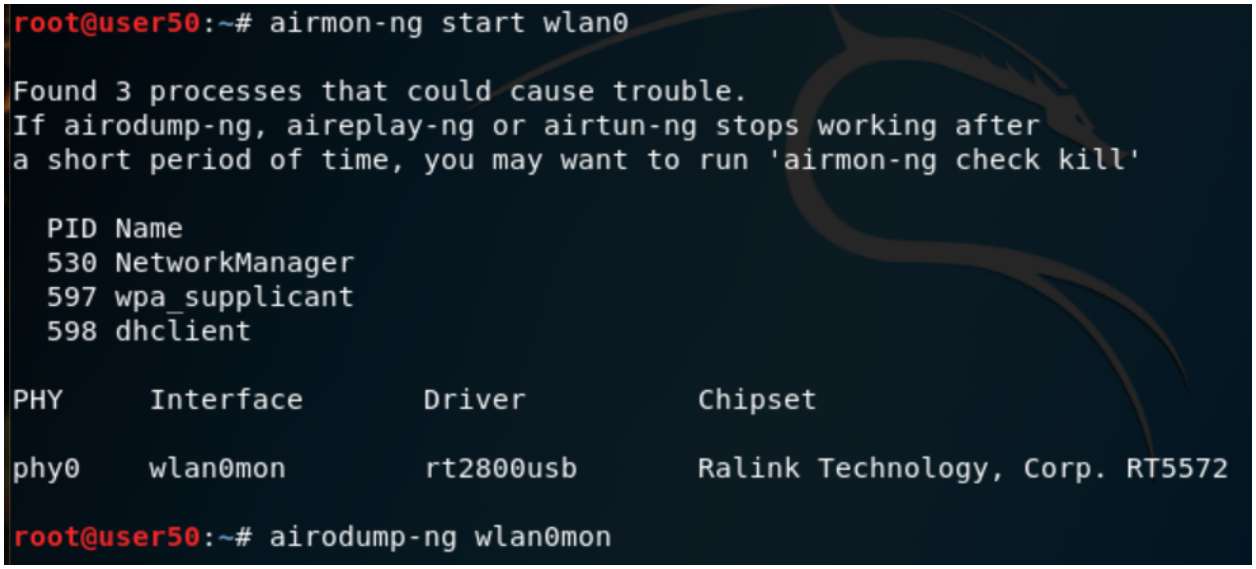
The first step that I did was to set up everything and make sure that I was running the required adapter for this project. I ssh'd into my dedicated machine using the command “ssh root@13.13.13.50” After gaining access, I made sure that the machine had the required adapter to exploit the access points as shown in figure 1.

Figure 1: Required Adapter



The second step that I did was to start the wlan0 and the command I typed was “airmon-ng start wlan0” as shown in figure 2. What I also did was to figure out if the access point of Rome was shown with the command “airodump-ng wlan0mon” also in figure 2.

Figure 2:



To find the manufacturer of the Rome access point, I entered the command “airodump-ng wlan0mon --manufacturer”. The --manufacturer is a command to find the manufacturer of the shown access points. This is shown in figure 3.

Figure 3:

Unfortunately, the manufacturer for the Rome AP was unknown so I decided to look up the bssid onto a website I found that turns the bssid into a oui lookup website. I took the bssid of Rome and inserted that into the oui search onto the wireshark website, in figure 4.

Figure 4

Examples:
0000.0c
08:00:20
01-00-0C-CC-CC-CC
00d9.d110.21f9
01-23-45-67-89-AB-CD-EF
missouri

OUI search

B0:BE:76:08:BE:0C

Find

Results

B0:BE:76 Tp-Link Technologies Co.,Ltd.

I then tried to see if there were any connections onto Rome AP and to see if there were enough data packets to collect to crack the hex key in the access point. It turns out that after a while of letting it run, I couldn't find enough data. In figure 5 and 6. Figure 5 shows the amount of time I let the machine scan the network and figure 6 is after I typed in the command "aircrack-ng <filename>" in this case it was rome-01.cap.

Figure 5:

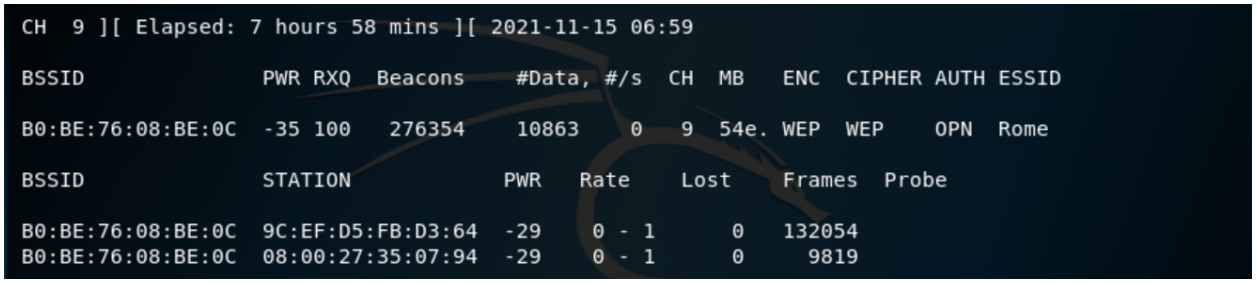
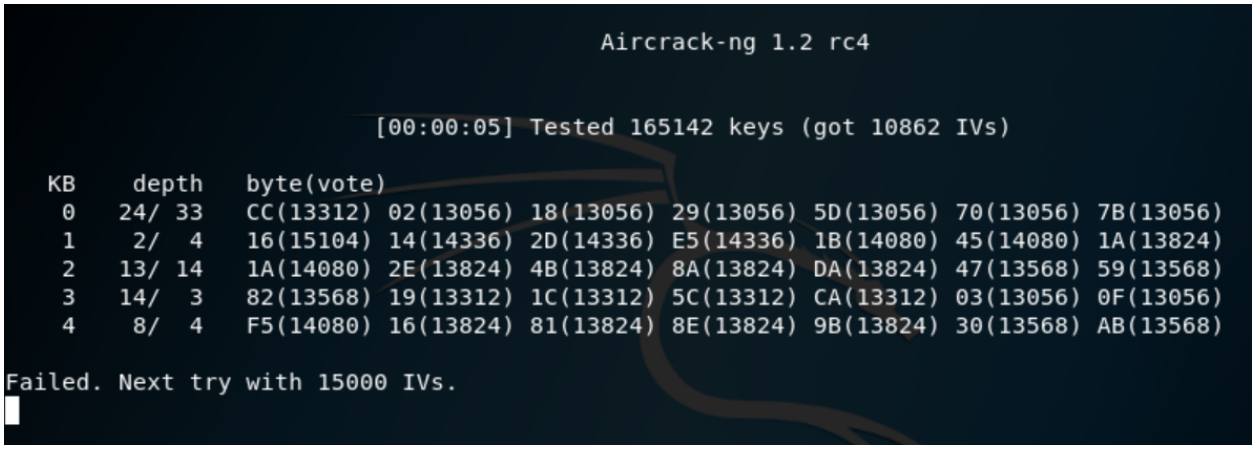
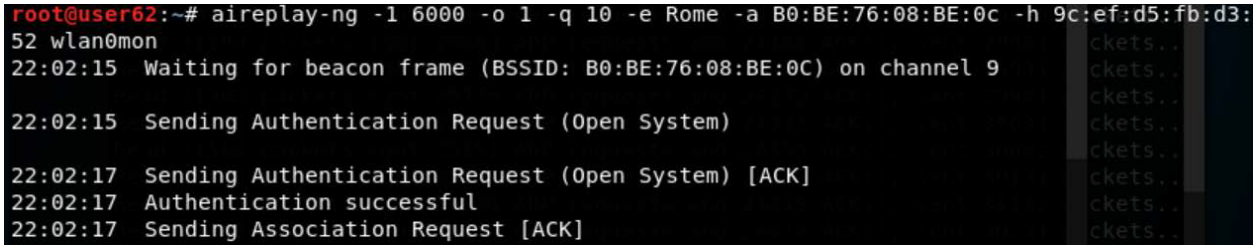


Figure 6:



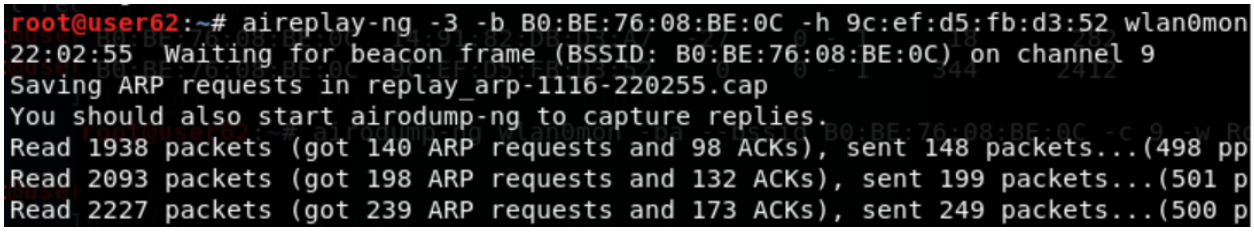
The next step was to figure out any other way that I can find this key. I then went onto the aircrack website and tried the steps for a simple wep crack. It turns out that the Rome AP did not have any connected clients and was losing packets. I then researched a way to create a fake connection to that AP. I found it on aircrack's website again with the commands as well. I went onto my system and typed in the command "aireplay-ng -1 6000 -o 1 -q 10 -e <access_point_name> -a <mac_address> -h <host_mac_address> wlan0mon". Figure 7 shows this command.

Figure 7:



I then went ahead and tried an arp request replay attack onto the system with the command, "aireplay-ng -3 -b B0:BE:76:08:BE:0C -h 9c:ef:d5:fb:d3:52 wlan0mon". Here this process sent out packets to the connection with that fake connection onto my host machine as shown in figure 8.

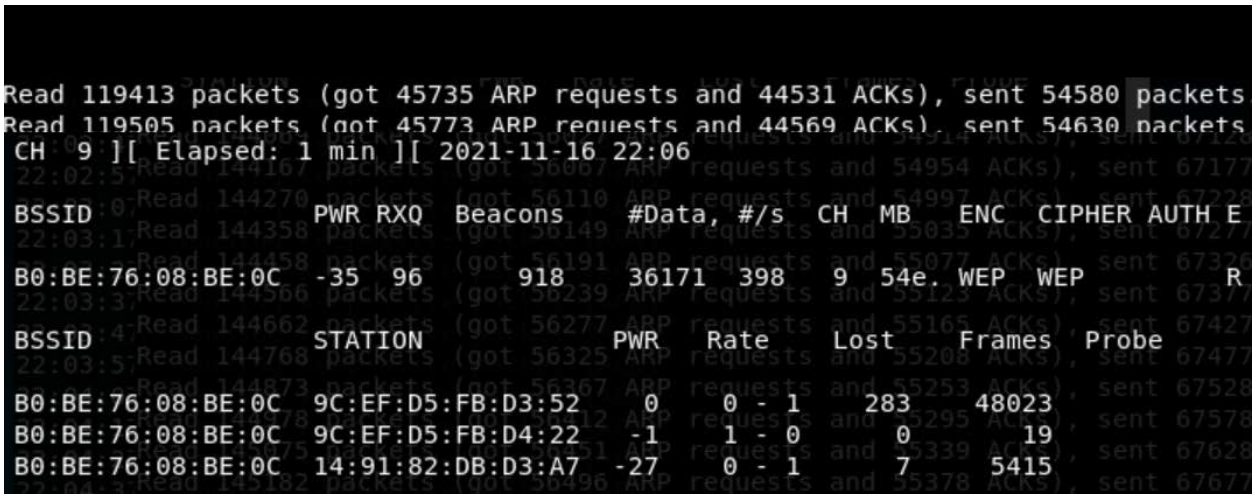
Figure 8:



Next step was to look and see if my fake connection was connected to the Rome AP so I typed in the command “airodump-ng wlan0mon -ba -bssid <rome_mac_#> -c 9 -w Rome”. This command went ahead and saved any packets that went through. Figure 9 and 10 shows the captured packets.

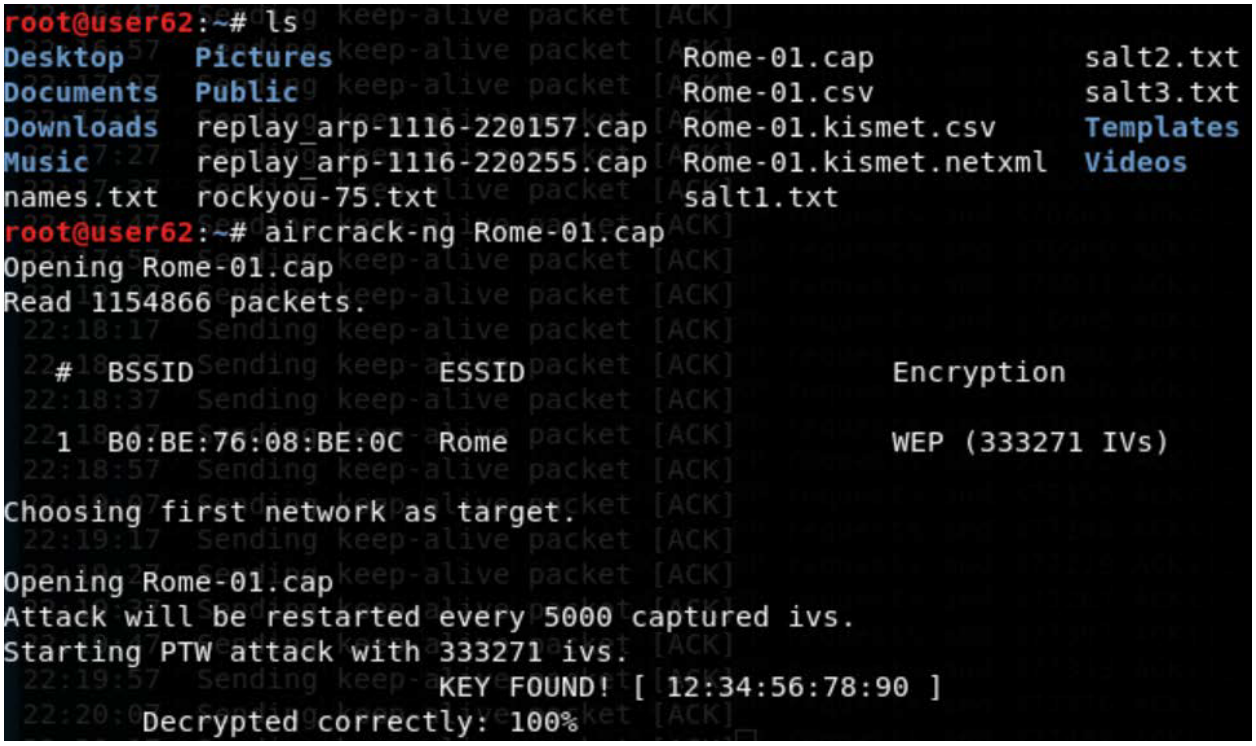
Figure 9:

Figure 10:



Once I had saved the captured packets, it was like project 3 part 1. I first decided to search for my recently captured file under the name of “Rome” in the root directory and I did find it. I then went ahead and cracked the cap file using the “aircrack-ng <file_name>” command as shown in figure 11. I then found the key to the Rome AP.

Figure 11:



1.7 A conclusion section discussing how easy/difficult your experience was

The second part of the project to find both the Miami and Rome aps were very hard compared to the first part. When finding the basic information for the APs such as the bssid, channel number, and manufacturer was pretty easy to find since that did not require a lot of knowledge to find. When it got to cracking the hex key and finding it, this is where the difficulty came. I read through the websites that the professor provided and also looked at the hints at first. I then tried create the fake authentication which was weird to understand along with the 3 ways to attack the access point. The first attack that I tried was the arp request attack and that did not work, next I tried the fragmentation attack and that did not work as well. This is where I started to look at other resources and to make sure that my addresses were the right addresses and the commands were typed right. I then got a breakthrough with the chopchop attack. I then learned from project 3 part 1 with taking time to gather the data. I let the program run and ran the various commands, packetforge along with aircrack. I then finally found the key. This was probably the hardest to find at first.

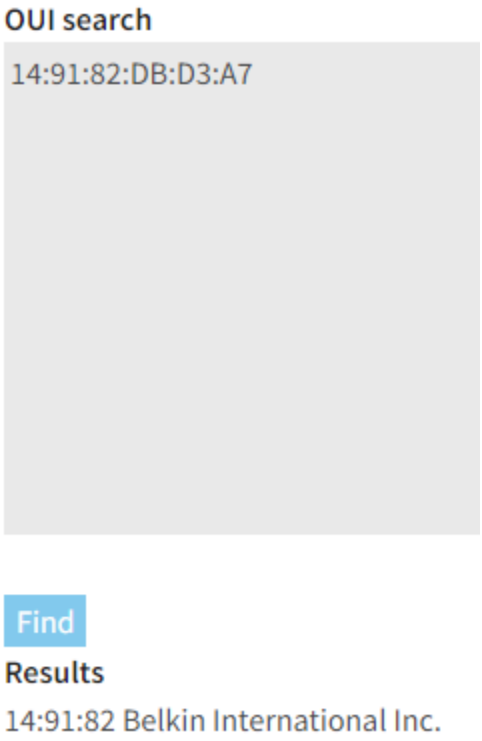
Section 2: Report on exploiting Miami AP

2.1 The bssid for Miami is **_14:91:82:DB:D3:A7_** .
2.2 The channel for Miami is **__6__** .
2.3 The manufacturer of Miami is **_Belkin International__** .
2.4 The HEX key (a.k.a. password) for Miami is **_12:12:12:12:12_** .
2.5 This attack took me/us **_5_** hours to perform.
2.6 Step-by-step documentation of how you performed the exploitation

For the Miami AP, I took the same first steps to figure out the bssid, channel number, manufacturer in the terminal, and starting the system onto monitor mode. To find that I typed in the command “airodump-ng wlan0mon” to find Miami’s bssid and channel number shown in Figure 1. To find the manufacturer I typed in the command “”airodump-ng wlan0mon --manufacturer”. I then I took the bssid address and plugged it into wireshark to find the manufacturer in Figure 2.

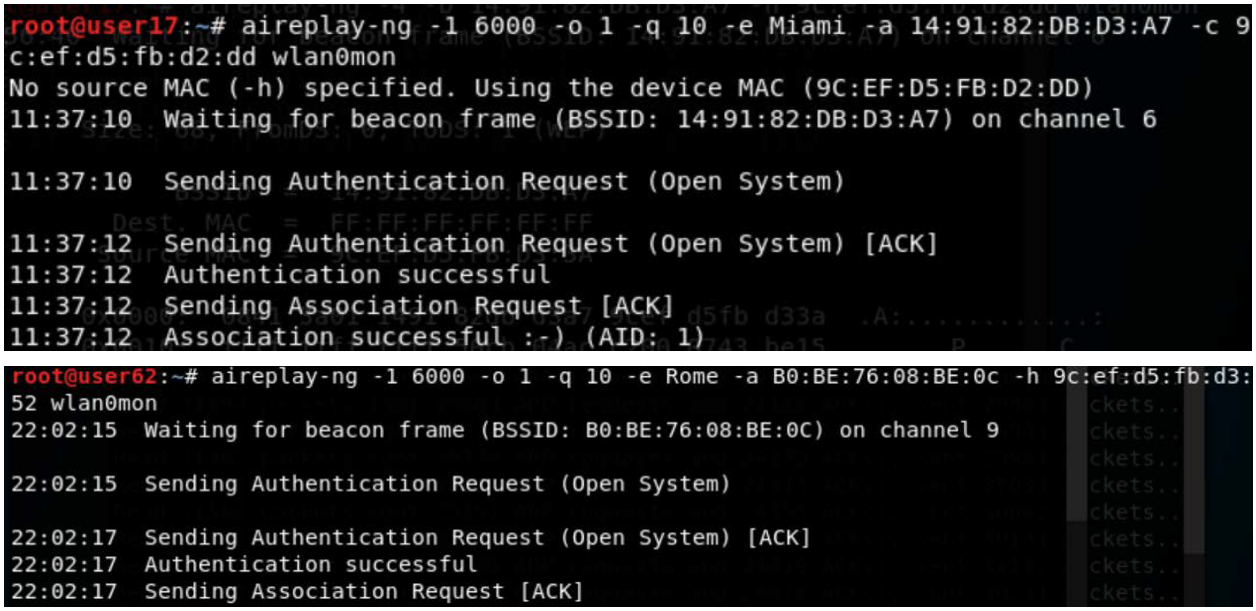
Figure 1: airodump-ng

Figure 2:



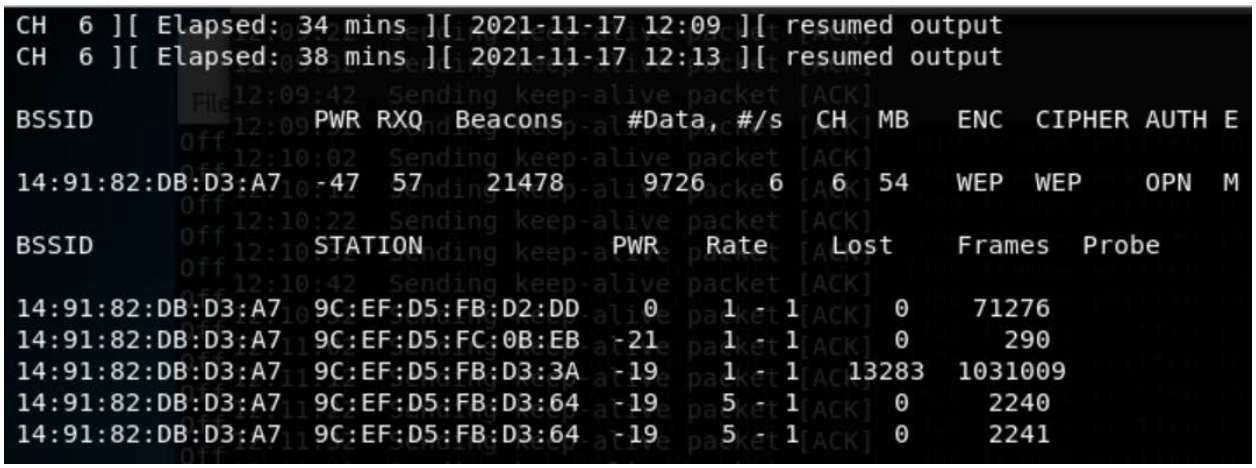
I then decided to create a fake connection onto Miami’s ap since the entire project is based on cracking passwords with no connected clients. To do so, I typed in the command “aireplay-ng -1 6000 -o 1 -q 10 -e Miami -a 14:91:82:DB:D3:A7 -c 9C:EF:D5:FB:D2:DD wlan0mon”. In the figure 3 below, I changed the command from “-c” to “-h” because it did not accept the source MAC but it still worked before and after I changed the command.

Figure 3:



Next up, I decided to follow the steps that I did on Rome's AP. I decided to make sure that my fake connection was up and running. I typed in the command "airodump-ng wlan0mon -ba --bssid 14:91:82:DB:D3:A7 -c 6 -w Miami". I had also decided to save any data packets that went through as well shown in figure 4.

Figure 4



After I knew it was connected, I decided to use the chopchop attack on Miami. To do the attack, I typed in the following command, "aireplay-ng -4 -b <Miami_mac#> -h <host_mac#> wlan0mon". This then prompted a yes or no question whether or not I wanted to use the selected packet. I then typed in "Y" to confirm that I wanted to use the packet. It then proceeded to do the chopchop attack. Figure 6 shows the results of the chopchop attack.

Figure 5:

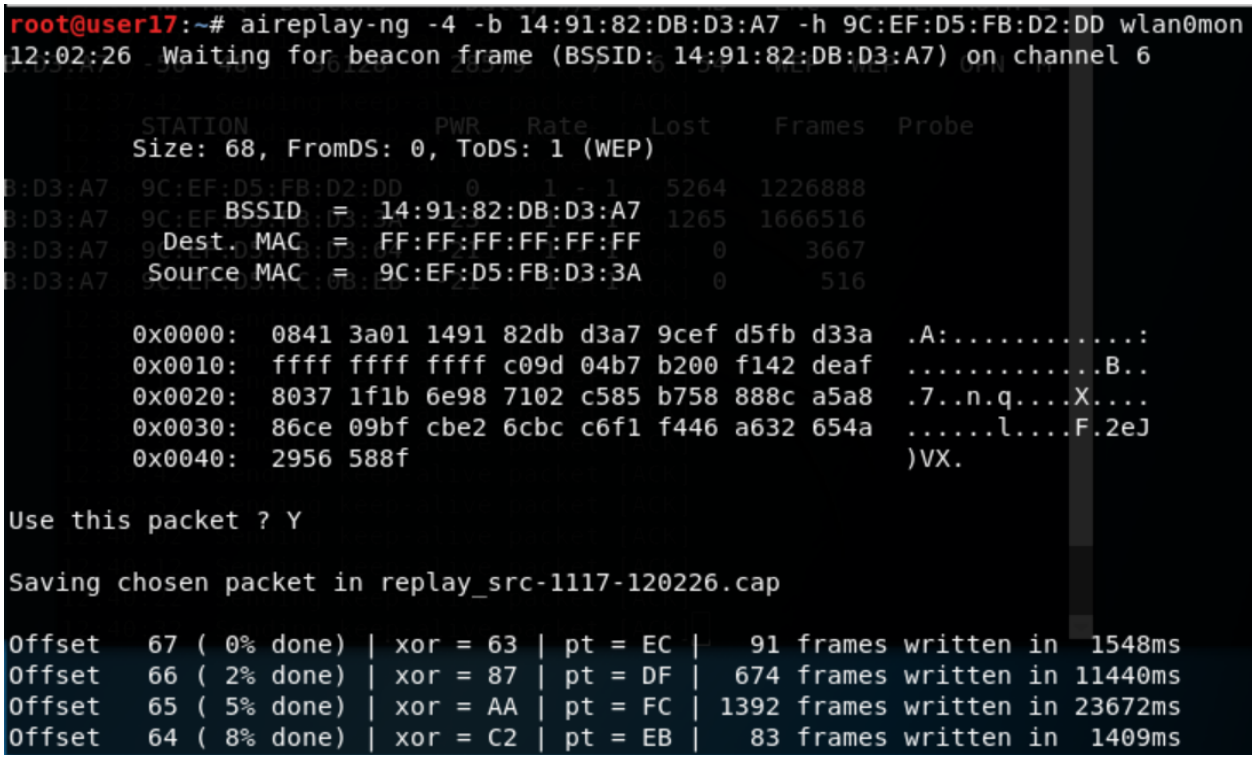
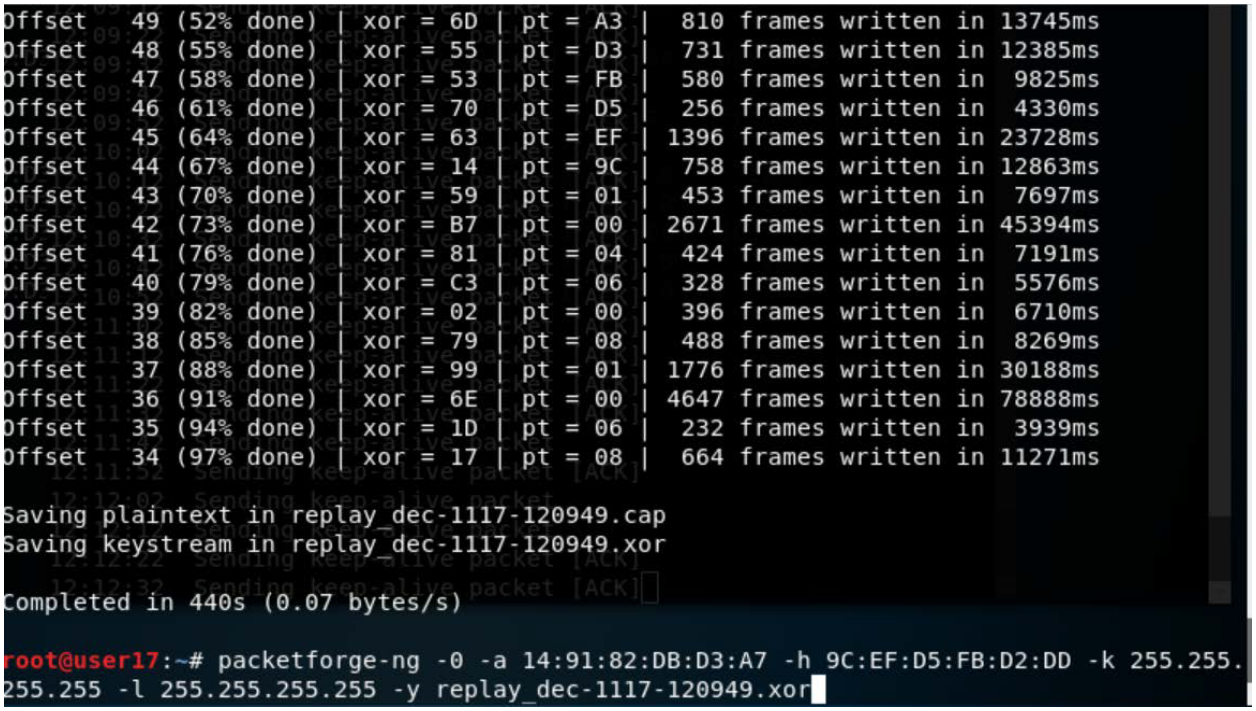
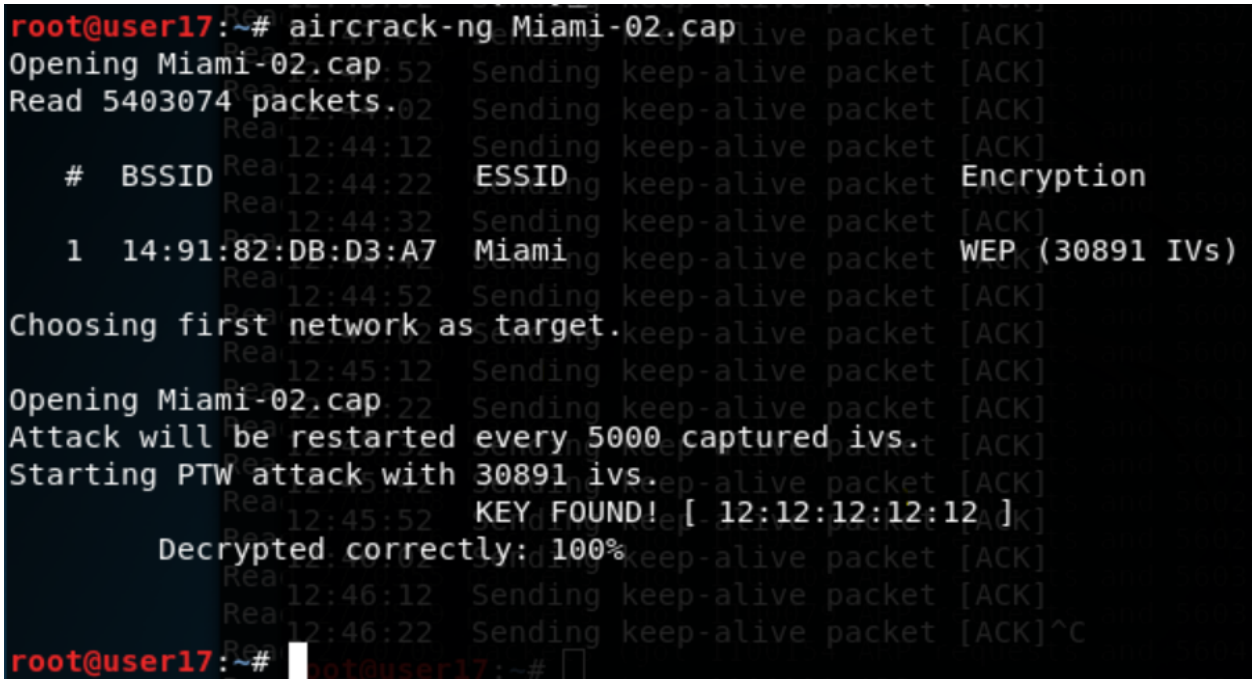


Figure 6:



Once packetforge command is done, I went to see my collected data that I gathered on my terminal. I then used the aircrack command to crack the cap file I gathered a while ago. To do it, I used the command, “aircrack-ng <filename>” in this case the file name I saved was Miami-02.cap and figure 7 shows this.

Figure 7.



2.7 A conclusion section discussing how easy/difficult your experience was

To find this password for the Miami ap was kinda difficult but easier compared to the rome AP. I went through the same process that I did with Rome. I went through each machine and made sure that the adapter was connected and turned into monitor mode. Once that was done, I went through found the bssid, manufacturer, and channel number. Next up was to find a way to create that authenticated connection to the AP. I did so and next was to try to see if that connection was up and running. At first, it was connected but no packets were being sent. I think that makes sense because I did not type in any commands to inject the access point with at this point. After that, I decided to use aireplay-ng -4 to do so like the Rome AP. I went ahead and found the password the same way that I did with the Rome AP. Overall the project was very difficult because my machines kept disconnecting and that I had to restart everything from collecting files to cracking everything. It was frustrating to get through and then also my chopchop attack didn’t work for the first few times I tried it. I eventually ran it enough times that it worked. I believe that this project was super hard to figure out and understand but once you crack one, then you could follow the same steps you did with the other APs.

Sources

https://www.aircrack-ng.org/doku.php?id=airodump-ng
https://www.aircrack-ng.org/doku.php?id=cracking_wpa
https://www.wireshark.org/tools/oui-lookup.html
https://beta.ivc.no/wiki/index.php/WEP_Cracking
https://www.aircrack-ng.org/doku.php?id=simple_wep_crack
https://www.aircrack-ng.org/doku.php?id=arp-request_reinjection
https://www.aircrack-ng.org/doku.php?id=how_to_crack_wep_with_no_clients
https://www.aircrack-ng.org/doku.php?id=packetforge-ng