|  |
| --- |
| Dark Wolf Solutions |
| Hack Our Drone  Module 3: Communications  Lab Manual |
|  |
| 2023 05 08 |

# 

# **Task 1 Describe the Comm System Under Test**:

## **Task 1 Objective**:

Examine the components of this UAS Communication system and understand its function.

## **Task 1 Description**:

To better understand the communication system under test, answer the following questions:

1. What type of data passes between the UAV and the GCS?
2. What type of communication link is used to pass that data?
3. What frequency or frequencies are used for this comm link?
4. What security protocols are used on this comm link?
5. Which component provides the Wireless Access Point (WAP)?

## **Task 1 Solution:**

### **A1: What type of data passes between the UAV and the GCS?**

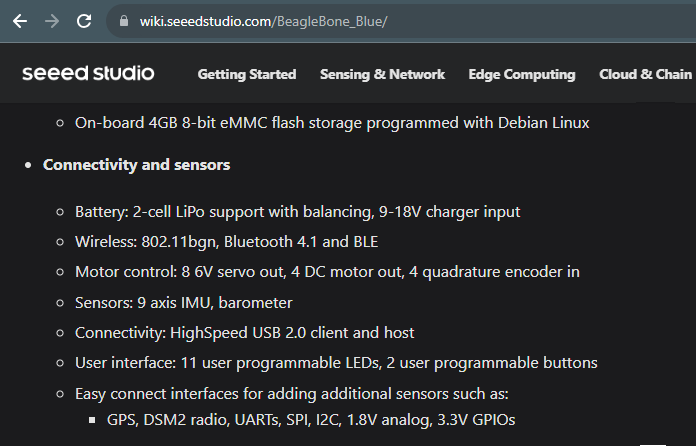
Typically, the type of UAS we are interested in run two types of data:

1. RC (radio controller) serial data describing the state of the gamepad controls used for direct flight control by a pilot
2. MAVLink protocol data which sends the current state of the UAV and receives commands and/or flight plans (from the point of view of the UAV).

In this configuration, there is no separate radio providing RC data. The gamepad controls are sent to the UAV as MAVLink data packets over the Wi-Fi connection.

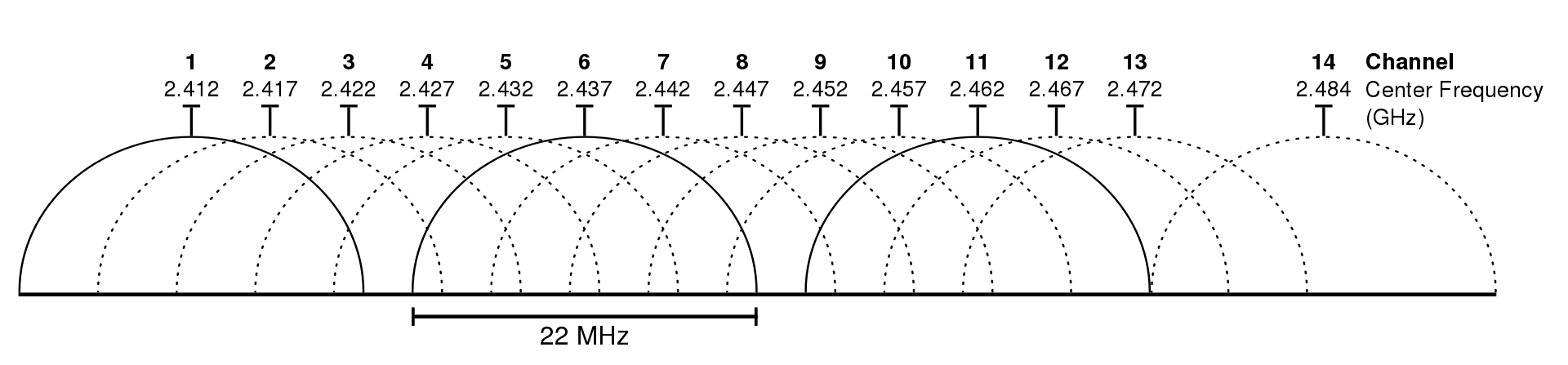
### **A2: What type of communication link is used to pass that data?**

From the specifications listed on the BeagleBone Blue website, there is on-board 802.11bgn wireless, Bluetooth, BLE, and optional radios. For this configuration, only the 802.11bgn (Wi-Fi) is being used for communication between the GCS and the UAV although other configurations are possible.



### **A3: What frequency or frequencies are used for this comm link?**

802.11b, 80211b, and 802.11n (and combinations thereof) all use a 2.5 GHz frequency. A fixed bandwidth around this frequency is subdivided into channels with a specific channel center frequency and channel bandwidth. In this US, channels 12 and 13 are only used under special restrictions while channel 14 is banned from use for Wi-Fi. While there are additional complexities around this topic, this is sufficient understanding for this lab.



<https://en.wikipedia.org/wiki/IEEE_802.11#/media/File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).svg>

In order to reduce interference between UAS, different UAS have been assigned to use different channels although they are not necessarily exclusive assignments.

### **A4: What security protocols are used on this comm link?**

(screen shot from phone)

### **A5: Which component provides the Wireless Access Point (WAP)?**

In this configuration, the UAV is providing the WAP to which the GCS Android phone is connecting.

# Task 2: Capture 802.11 WiFi Traffic

## Task 2 Objective:

We want to ‘sniff’ the network traffic between the GCS and the UAV. Captured network traffic can be analyzed for security issues.

## Task 2 Description:

It is possible to ‘eavesdrop’ and to WiFi traffic as a third-party without connecting to a wireless hotspot. We do this using a WiFi device that has a wifi chip that can be set to special mode known as ‘managed.’ Normally, the network devices and software on our computers will ‘ignore’ traffic not intended for our device. But in managed mode, we sniff the traffic of other users and devices.

## Task 2 Solution:

### Files:

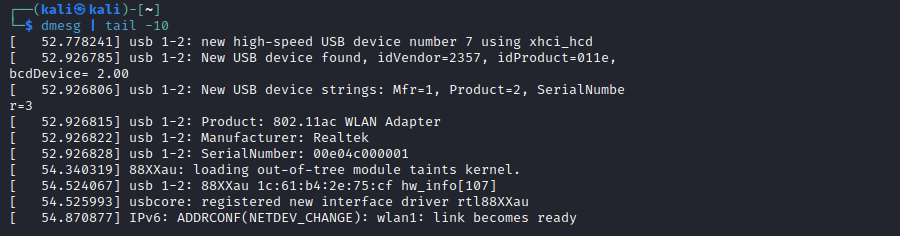
If you run into trouble capturing the files as described here, you can still examine the wifi capture files located in “Labs > 03-comms > Files”

### Setup:

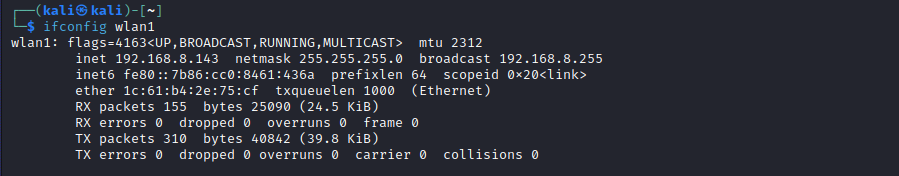
1. Power up the UAV
2. Connect the GCS Android WiFi to the UAV Hotspot (e.g Airwolf-A123)  
   

### Solution:

1. Boot up a Kali laptop
   1. Boot into “Live Persistent' mode
2. Login as kali
3. Insert the TP-Link T2U Nano USB WIFi Dongle into a USB port
4. In a terminal, run the command dmesg to verify that the WiFi chip was seen by the operating system. The output should be similar to this



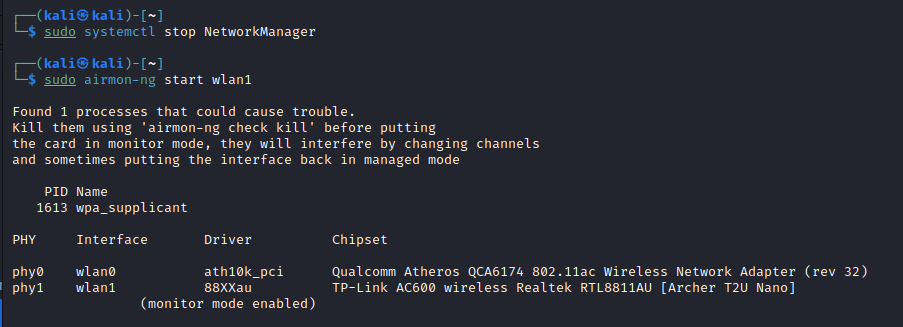
1. Note the final line in the block above. It defines the device name for the network interface. In this case, it is wlan1
2. Using the device name determined above, run ifconfig wlan1 to view the network interface. The output should be similar to this



1. Run the following commands to put the network interface into ‘monitor’ mode

sudo systemctl stop NetworkManager

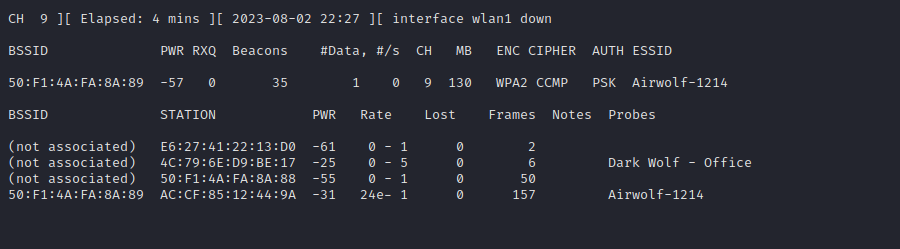
sudo airmon-ng start wlan1



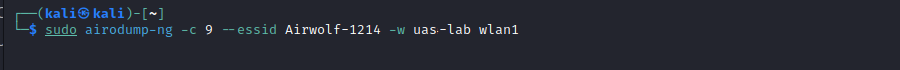
1. Start sniffing the network using the airodump-ng command. But tune it to just listen to your UAS Wi-Fi matching the last four hex characters to that on ***your*** UAS
   1. sudo airodump-ng --essid Airwolf-1214 wlan1



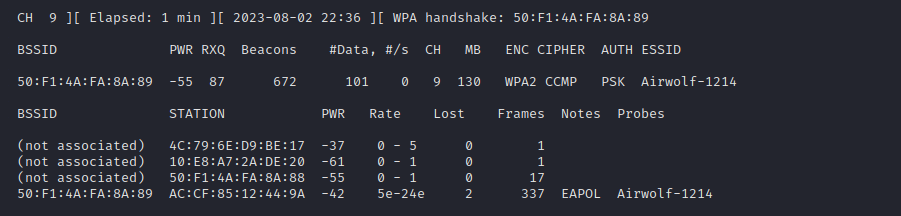
1. In the top part of the output, wait for the SSID that matches your UAS (*Airwolf-1214 in this example* but where the last four characters match those on ***your*** UAV). When you see it, hit ‘q’ twice to exit airodump-ng. There are note the following two items;
   1. The ESSID which will be matches the Wi-Fi of your drone
      1. In this case it is Airwolf-1214
   2. The Channel number CH.
      1. In this case 9, your channel number may be different
      2. Note this channel number for use in the next step



1. Now we can narrow our packet capture to just the network traffic associated with our UAS. Run the following command to capture traffic into a set of files with the label ‘*uas-lab-01*’ *(if you run it more than once, the number will increment).*h Use the last four hex characters in the ESSID that match ***your*** UAS. Use the channel number noted in the previous step to set the ‘-c’ flag
   1. airodump-ng -c 9 --essid Airwolf-1214 -w uas-lab wlan1



1. Let the packet capture run for a minute or more while you turn the wifi on and off on the Android phone. Watch for the ‘WPA Handshake’ in the top right hand corner of the terminal. When you see it, enter ‘*q*’ twice to end the capture.



1. Run the command ls -1 and you should see the five files you just created
   1. uas-lab-01.cap
   2. uas-lab-01.csv
   3. uas-lab-01.kismet.csv
   4. uas-lab-01.kismet.netxml
   5. uas-lab-01.log.csv
2. Use Wireshark to view the packet capture by running *wireshark* in the terminal or starting it from the menu. Use File > Open to select the packet capture file uas-lab-01.cap.
   1. wireshark
3. In this screenshot, you can see the following packets related to WPA2 authentication. You should be able to see these same elements with different packet numbers in your own capture file.
   1. An Association Request and Association Response in packets 15 and 17.
   2. Three (3) of the four (4) components of the ‘four-way handshake’ in packets 19, 21, and 23. The missing component is likely just a missed packet because collection isn’t alway perfect.

# Task 3: Crack the WPA2 Passphrase

## Task 3 Objective:

In this task, we will crack the WPA2 passphrase used to connect the UAV Wi-Fi Hotspot. (Wireless Access Point - WAP)/

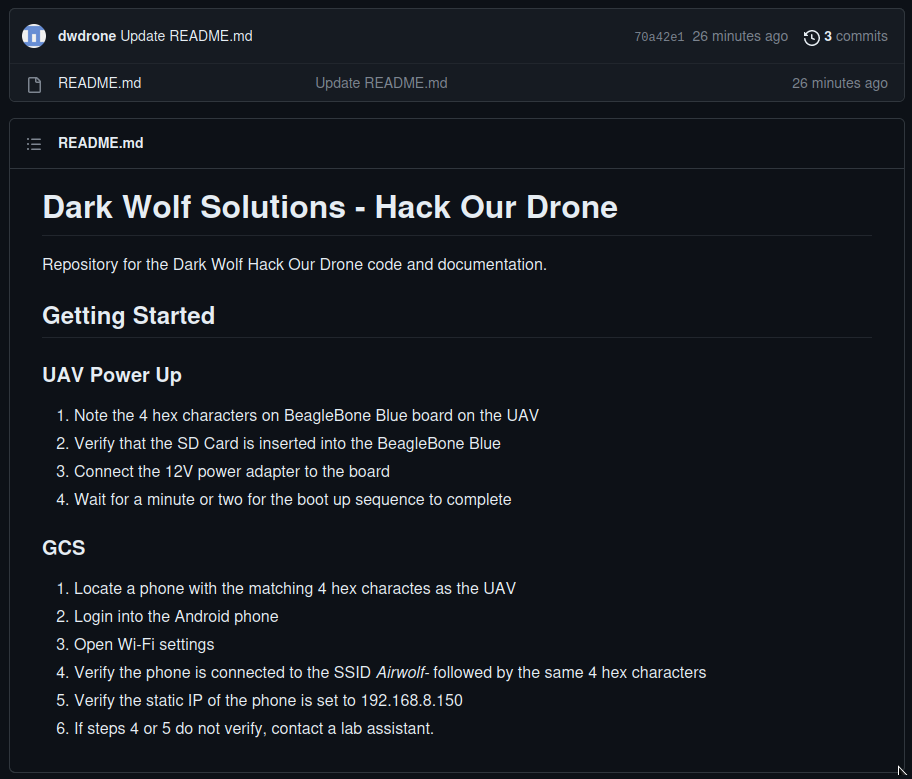
## Task 3 Description:

Now that we have captured a WPA2 authentication handshake, we can attempt to crack it with a dictionary attack. Lists of commonly used passwords have been compiled from passwords leaked in data breaches. There are over 750 million leaked passwords at this time.

But what happens if a UAS vendor is smart enough to avoid commonly used passwords? Is there any way we can still crack the WPA encryption? There are multiple ways to crack WPA depending on the exact Wi-Fi gear and protocols being used, but it can be difficult. In this task, we are going to attempt a dictionary attack using a password list based on key words we scrape off a UAS website. Since the lab is likely isolated from the internet, we will just create a mini web server on your lab computer and use a small web page that contains key words related to our UAS.

Due to available lab time, we have deliberately created a small wordlist that will quickly crack the password in the Wi-Fi capture. But our testers have used very similar techniques to these to successfully crack commercial vendor passwords that were based on company or product names, company addresses or phone numbers, and similar published information.

If you do have internet access, you can view this file at <https://github.com/dwdrone/lab>

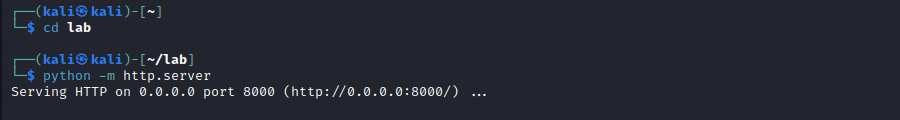


## Task 2 Solution:

### Files:

### Setup:

1. Open a terminal on Kali and cd into the ‘lab’ directory.
   1. cd lab
2. Start a mini web server using the following command
   1. python -m http.server



### Solution:

1. Run the following command to have the program *cewl* read the web page, parse the words from it, and store it in a new file.
   1. http://localhost:8000



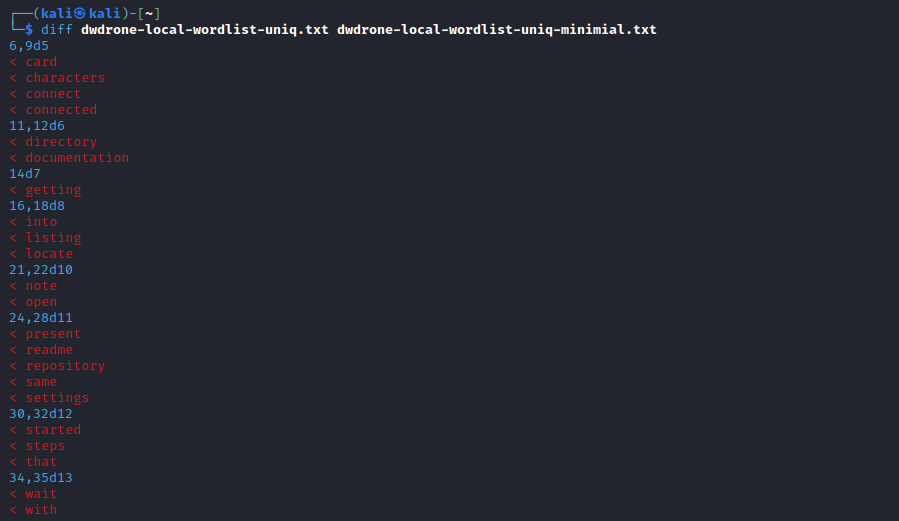
1. Sort this file and remove all the duplicates using the following command.
   1. aircrack-ng -w wifi-dictionary.txt wifi-01.cap



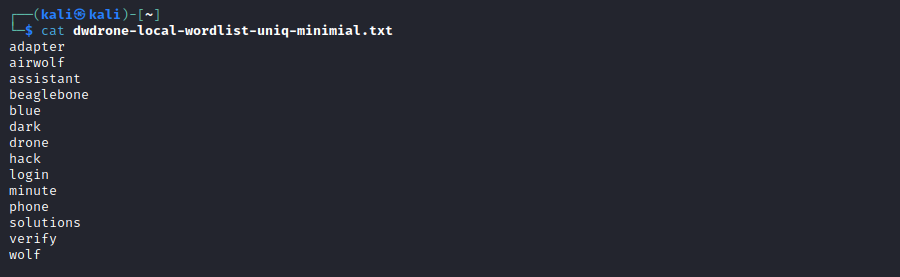
1. Copy this file to a new one in which we will make manual edits.



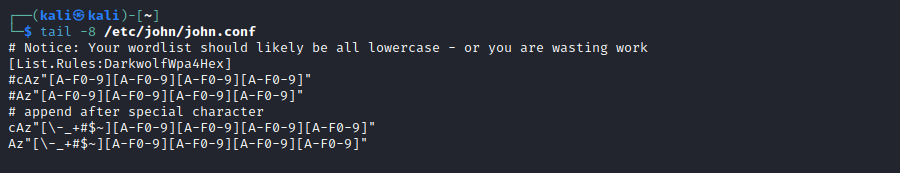
1. Manual edit the file to remove common verbs, prepositions, and overly generic nouns. The remaining list should include mostly nouns that might be related to the UAS and would seem to make a good base word for a password.
   1. Here are the words we deleted



* 1. Here are the words we kept

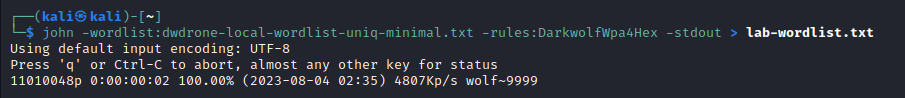


1. Next, we will feed this wordlist into a password generator. The password generator ruleset DarkwolfWpa4Hex is already written into the /etc/john/john.conf file. You can view it by running the following command
   1. tail -8 /etc/john/john.conf

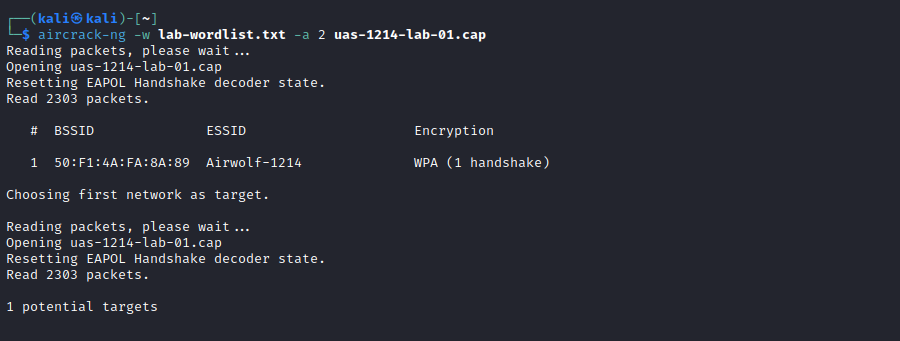


1. Generate the password list using the ‘John the Ripper’, the list of words, and the ruleset described above. The generated password list has 11 million entries.
   1. john -wordlist:dwdrone-local-wordlist-uniq-minimal.txt \

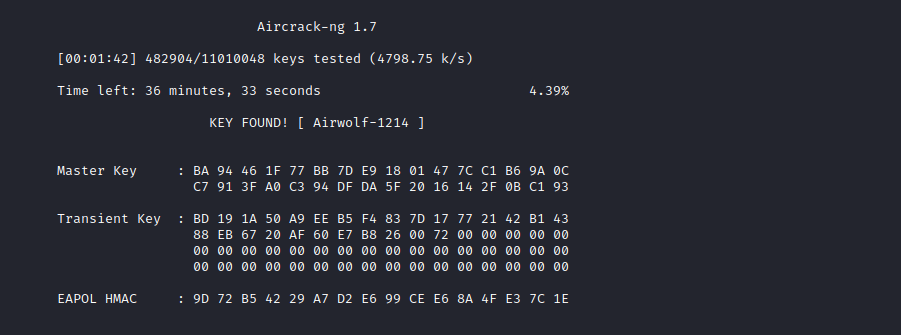
-rules:DarkwolfWpa4Hex -stdout > lab-wordlist.txt



1. Lastly, use aircrack-ng to find the Wi-Fi password. Note that to use ***your*** packet capture, use the file uas-lab-01.cap
   1. aircrack-ng -w lab-wordlist.txt -a 2 uas-lab-01.cap



1. The passphrase for the UAS wifi should be found in less than five minutes.



# Task 4: View the UAS MAVLink Traffic

## Task 4 Objective:

In this task, we will capture and review MAVLink Traffic broadcast from the UAV.

## Task 4 Description:

In this configuration, the UAV is only broadcasting to a single fixed IPv4 address which is 192.168.8.150. We will configure a network interface on the Kali box to listen to that address, capture MAVLink traffic, and decode it using the Wireshark MAVLink Protocol Decoder Plugin.

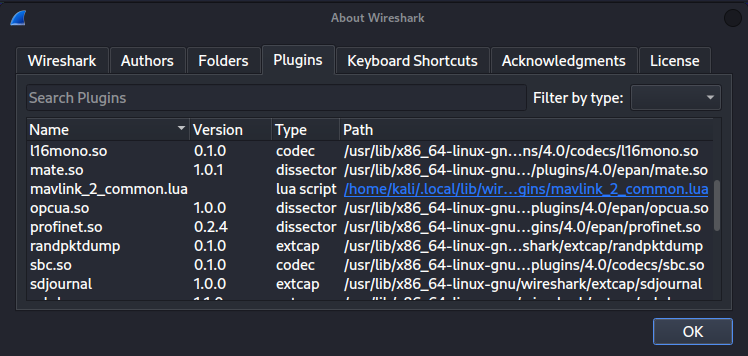
## Task 4 Solution:

### Files:

If you run into trouble capturing the files as described here, you can still examine the wifi capture files located in “Labs > 03-comms > Files”

### Setup:

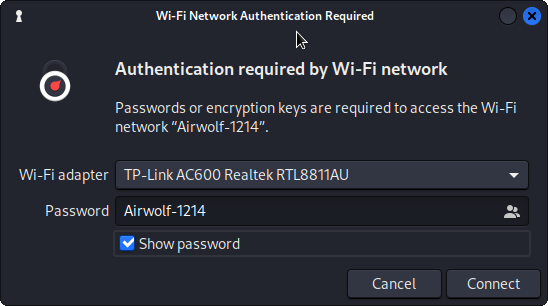
1. Connect the UAV and the GCS
   1. Power up the UAV
   2. Connect the GCS to the *Airwolf-A12*3 network
      1. Change the last four hex characters to match your UAS
   3. Start QGroundControl
   4. Verify that it connects to the UAV
2. Verify that the Wireshark MAVLink Plugin is installed
   1. Start Wireshark
   2. Select ‘About’
   3. Select the ‘Plugins’ tab
   4. Verify that MAVLink is listed
   5. If it is not installed, you can install it yourself
      1. mkdir -p ~/.local/lib/wireshark/plugins
      2. cp mavlink\_2\_commona.lua to ~/.local/lib/wireshark/plugins

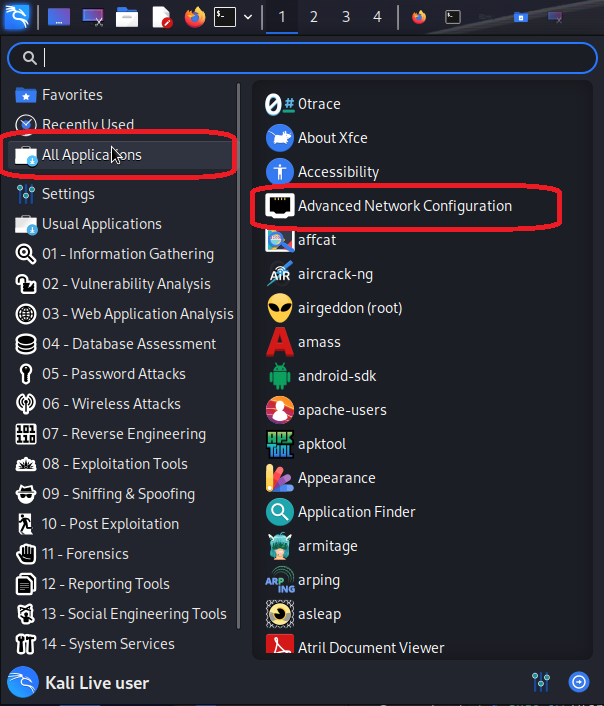


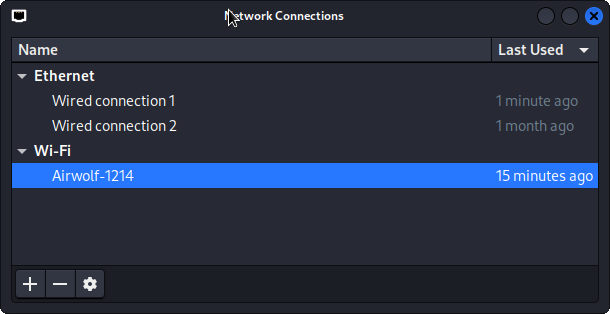
### Solution:

1. Insert the TP-Link T2U Nano USB Wi-Fi dongle to the laptop.
2. From the Wireless Networks in the upper right hand corner of the screen,

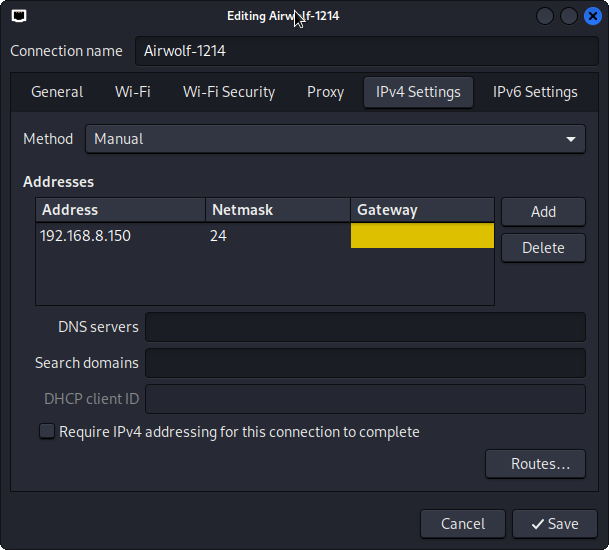
Select the “RTL8812AU” device and connect to the “Airwolf-A123” network, replacing the last four characters with those for your UAS. Enter the Wi-Fi password for ***your*** network.



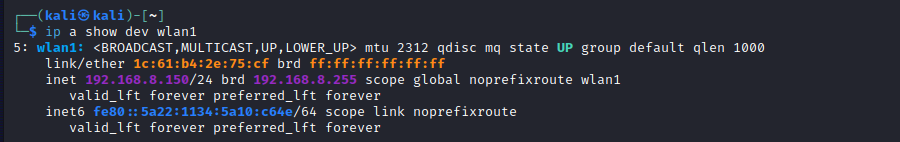
1. Open the Advanced Network Configuration from the application menu on the upper left corner of the screen.  
   
2. Select the *Airwolf-xxxx* SSID that matches your UAS.   
   Select the gear icon in the lower left to edit the configuration.



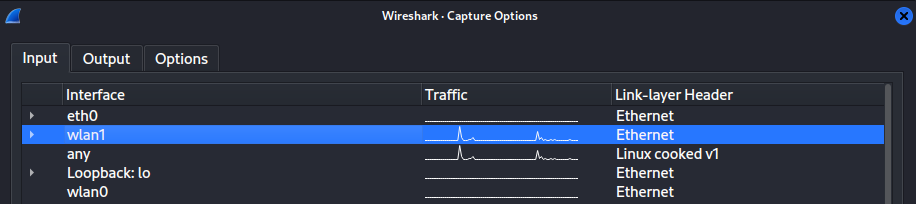
1. Configure the interface to use the
   1. a ‘Manual’ method
   2. a static IP address of 192.168.8.150
   3. a netmask of 24
   4. The gateway can remain blank
   5. Save



1. Disconnect the wireless interface and reconnect to reset the IP address.
2. Verify the interface is configured correctly on the command line



1. Start Wireshark and select the wlan1 interface



1. If you do not see packets flowing, you can reset the network interface on the Kali box -or- stop and restart QGroundControl on the GCS -or- both
2. Once you capture some packets, you will be able to decode most of them. Some packets generated in this version of the flight software are not correctly decoded in by the Wireshark plugin.
3. While it is likely that the flight software is not fully configured on your lab UAS, you can still see packet updates by selecting the “RAW\_IMU” packets and moving your UAV around a bit.

