

Tello Drone Exploration Lab

1 Overview

The goal of this lab is to become familiar with the drone controls, its network settings, and the ports it uses to send and receive information. This lab will introduce the DJITelloPy Python library as a method of controlling the drone.

2 Environment Setup

Install the Tello and X-Hubsan apps on your mobile device and DJITelloPy Python library on your computer. The DJITelloPy library is an open-source project and can be found here:

<https://github.com/damiafuentes/DJITelloPy>.

You can install it from source or use the command

```
pip install djitellopy
```

Our testing found that installing the library using pip was easily done on **Ubuntu 20.04**, but prerequisite issues occurred when attempting to install it on Ubuntu 16.04.

So you need to:

- Install Tello and X-Hubsan app (you can also use Bluestacks android emulator

<https://bluestacks.com>)

- Install DJITelloPy Python library

3 Drone Network Exploration

3.1 Network Access Settings

Connect to the drone using your Ubuntu 20.04 VM and run Wireshark to capture the traffic. You can use the sample code on GitHub:

<https://github.com/damiafuentes/DJITelloPy/blob/master/examples/simple.py>.

Q 3.1.1 [2.5 pts] What is the IP address of the drone?

192.168.10.1

Q 3.1.2 [2.5 pts] What is the IP address of your connecting device?

192.168.10.2

Include a screenshot from your computer or mobile device's network settings.

314	74.158081	192.168.10.1	192.168.10.2	UDP
315	74.257757	192.168.10.1	192.168.10.2	UDP
316	74.319209	192.168.10.2	192.168.10.1	UDP
317	74.360203	192.168.10.1	192.168.10.2	UDP
318	74.462136	192.168.10.1	192.168.10.2	UDP
319	74.565280	192.168.10.1	192.168.10.2	UDP

Q 3.1.3 [5 pts] What wireless security protocol is in use?

UDP

- **Q 3.1.4 [5 pts]** On the controller, which port is used to send commands to the drone?
 - 8889
- **Q 3.1.5 [5 pts]** On the drone, which port receives commands from the controller?
 - 8890

192.168.10.1	192.168.10.2	UDP	176 8889 → 8890 Len=
192.168.10.1	192.168.10.2	UDP	176 8889 → 8890 Len=
192.168.10.1	192.168.10.2	UDP	175 8889 → 8890 Len=
192.168.10.1	192.168.10.2	UDP	176 8889 → 8890 Len=
192.168.10.1	192.168.10.2	UDP	177 8889 → 8890 Len=
192.168.10.1	192.168.10.2	UDP	176 8889 → 8890 Len=

3.2 Important Ports

For this next step, you will need to control the drone while sniffing network traffic. The best way to do this is to run Wireshark on the device acting as the drone's controller. Connect to the drone on your computer and use the DJITelloPy library to create a controller for the drone on your computer. You can use the sample code on GitHub as a template in designing your drone controller: <https://github.com/damiafuentes/DJITelloPy/blob/master/examples/manual-control-opencv.py>.

Run the code while sniffing traffic and give the drone a few commands. View the packet capture and answer the following questions:

- **Q 3.2.1a [5 pts]** On the controller, which port is used to send commands to the drone?

8889

- **Q 3.2.1b [5 pts]** On the drone, which port receives commands from the controller?

8890

.1	192.168.10.2	UDP	176 8889 → 8890 Len=132
.1	192.168.10.2	UDP	174 8889 → 8890 Len=130
.1	192.168.10.2	UDP	177 8889 → 8890 Len=133
.1	192.168.10.2	UDP	175 8889 → 8890 Len=131
.1	192.168.10.2	UDP	176 8889 → 8890 Len=132
.1	192.168.10.2	UDP	176 8889 → 8890 Len=132

- **Q 3.2.2a [5 pts]** On the drone, which port is used to send video feed to the controller?

62513

- **Q 3.2.2b [5 pts]** On the controller, which port receives video feed?

11111

192.168.10.1	192.168.10.2	UDP	175 8889 → 8890 Len=1
192.168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=
192.168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=
192.168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=
192.168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=
192.168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=
192.168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=
192.168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=

- **Q 3.2.3a [5 pts]** On the drone, which port sends directional status messages to the controller?

8889

- **Q 3.2.3b [5 pts]** On the controller, which port receives the status updates from the drone?

8890

168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=1460
168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=1460
168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=1460
168.10.1	192.168.10.2	UDP	171 8889 → 8890 Len=127
168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=1460
168.10.1	192.168.10.2	UDP	1504 62513 → 11111 Len=1460

Include screenshots of Wireshark packets for each pair of these questions.

4 Drone Photo and Video Feed

4.1 Storage

Take a photo using this code:

<https://github.com/damiafuentes/DJITelloPy/blob/master/examples/take-picture.py>

Q 4.1 [5 pts] Was the photo that was taken stored on the drone or only on the VM? which port is used to send that photo to the controller?

On the VM. It is sent to 62513

4.2 Data Streams

Connect to the drone on your computer. Using the DJITelloPy library, fly the drone and capture video from your computer using this code:

<https://github.com/damiafuentes/DJITelloPy/blob/master/examples/record-video.py>


While you fly the drone, capture the raw data of the video stream using Wireshark. In Wireshark, this can be done by right clicking on a packet that is part of the stream and clicking Follow > UDP Stream. A window with the data stream should pop up. Change the “Show data as” field to “Raw” and click “Save As.”

Q 4.2.1 [10 pts] What file format are videos saved in and what is the video codec of the stream?

Convert the video stream to a format that your device can easily view. (Please do this with a command line tool rather than sketchy online services). Include a screenshot of the command used.

The video is saved in binary file and I converted it to mp4 video format by using command line tool below.

Video Codec.



Name:

stream

Type:

Binary (application/octet-stre...

Size:

47.7 MB (47,735,927 bytes)

Parent folder:

/home/amirmansha/Downloads

Accessed:

Thu 18 Nov 2021 03:10:07 PM EST

Modified:

Thu 18 Nov 2021 03:10:08 PM EST

I used ffmpeg command to convert the stream to video format.

```
amirmansha@amir-VirtualBox:~$ ffmpeg -i /home/amirmansha/Downloads/stream video.mp4
ffmpeg version 4.2.4-1ubuntu0.1 Copyright (c) 2000-2020 the FFmpeg developers
built with gcc 9 (Ubuntu 9.3.0-10ubuntu2)
```

Video

Dimensions: 960 × 720
Codec: H.264 (High Profile)
Frame rate: 25.00 frames per second
Bit rate: 926 kbps

Q 4.2.2 [10 pts] Attach a segment of this video (no longer than 15 seconds please) from the drone in your submission on Blackboard.

4.3 Capture the Flag

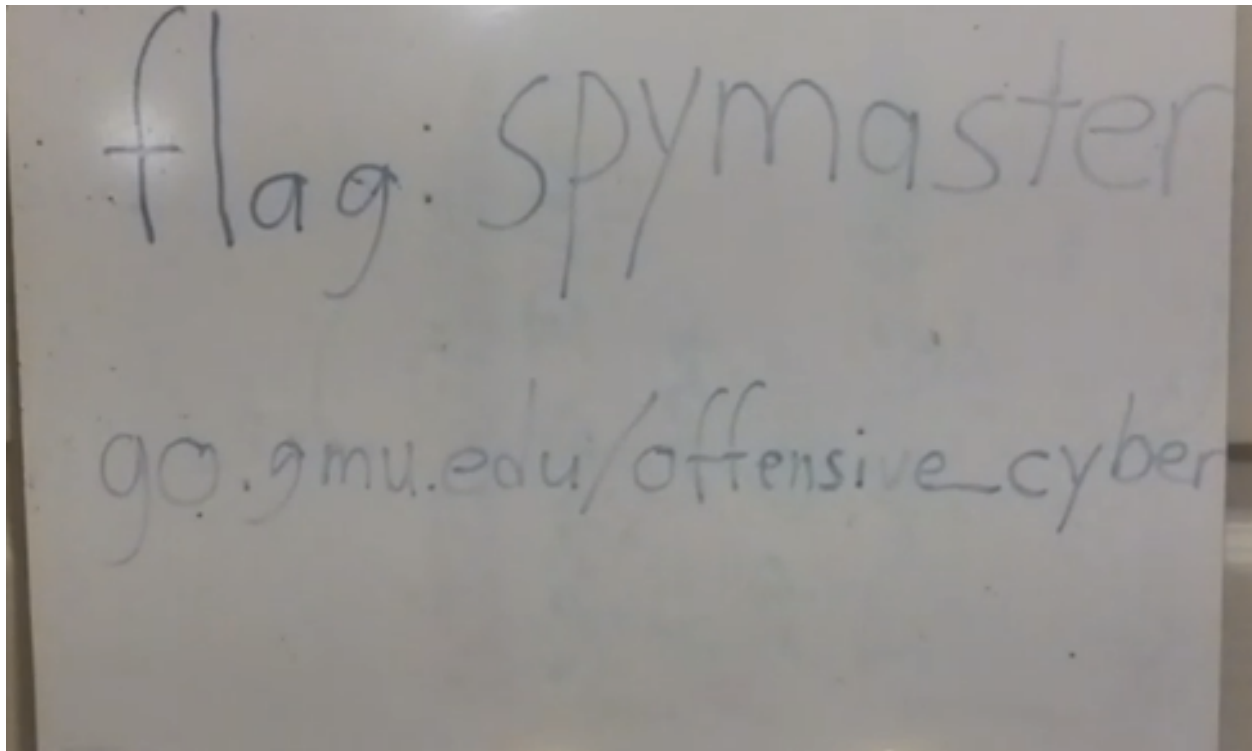
Download [this pcap](#) of drone traffic and extract the video from it.

Q 4.3 [20 pts] What is the flag shown in the video? Show screenshots and explain your steps.

Link to download:

<https://drive.google.com/file/d/1AAOKCqye6pSNtEq5NHbhpX--r1R-6qFQ/view?usp=sharing>

The flag shows said spymaster



```
amirmansha@amir-VirtualBox:~$ ffmpeg -i /home/amirmansha/Downloads/ctfc ctf.mp4
ffmpeg version 4.2.4-1ubuntu0.1 Copyright (c) 2000-2020 the FFmpeg developers
built with gcc 9 (Ubuntu 9.3.0-10ubuntu2)
configuration: --prefix=/usr --extra-version=1ubuntu0.1 --toolchain=hardened
```

I chose a packet from the ctf.pcap file and saved it as raw udp stream format then I used the command above to convert it to video format.

5 Drone Commands

View the data stream containing the command and control traffic sent to and from the drone.

Include a screenshot of the data stream.

Q 5.1 [5 pts] What keyword initiates the control mode?

Command

Include a screenshot of command and control traffic.

Q 5.2 [5 pts] What two possible responses can the drone reply with when receiving a command?

Ok and error

6 Submission

Submit a lab report answering the questions in this lab and providing labeled screenshots where applicable on Blackboard. Screenshots should provide comments about what is happening in the image. Attach a unique video extracted from section 4.2 (no longer than 15 seconds please) from the drone in your submission on Blackboard as well.