COSC235 - Homework 3 (Drones!)[*]

Assigned April 4th, 2016; Due 12:30pm on April 28th, 2016

Prof. Micah Sherr

Note: This homework assignment is worth 145 points (170 points with maximal extra credit awarded). This is a group assignment that should be completed in instructor-approved groups of 4-5 students (see below). Importantly, please note that there are deliverables that are due before the April 28th deadline.

1 Drones! {85 points}

For this assignment, you will be working in groups of 4-5 students. All group members will receive the same grade for this portion of the assignment, regardless of individual contributions or the size of the group. (Please be responsible group members!) **You have until Friday, April 8th at 8pm EDT to post the names of your group members and a (non-controversial) team name for your group to Piazza.** (You can make the post private to the teaching staff.) Otherwise, we will assign you to a group and make up a team name.

You will be working with Parrot AR.Drone2.0 Elite Editions. These cost a wee bit less than the drones used for military operations, but they are still relatively expensive.

Some notes about safety: (READ THESE CAREFULLY!!!)

- The drones are not weaponized. Do not turn them into weapons.
- The rotors (blades) spin very fast. I haven't personally stuck my finger in them, and I doubt it would do much damage (to a human finger), but it would probably still hurt. PLEASE BE CAREFUL. Make sure that you are a safe distance from the drone at all times. If the drone is too close to *any* person or living thing, **immediately** kill the drone's power. Do not attempt to recover. We will value living things far more than the drone.
- Always use the larger cover that covers all four rotors.
- From the user guide:

> *Avoid using the AR.Drone around domestic animals, particularly pet dogs. The AR.Drone incorporates an ultrasound altimeter (emission frequency: 40kHz). Ultrasonic waves are*

[*] Last revised on April 4, 2016.

> *known to disturb certain domestic animals, most especially dogs. Using the AR.Drone in the company of a dog may lead to it behaving erratically or perhaps even dangerously.*
>
> So let's avoid using it around dogs, or really any non-human animals.

Safety comes first. But the drones are expensive. Please try to be careful:

- **IMPORTANT NOTE:** You may only fly the drones indoors. Flying these drone outside is not legal — there are laws against operating drones around the D.C.-Metro area. I repeat, if you are outdoors, you may not, under any circumstances, operate the drone.
- Always implement an emergency mode that, if activated, immediately kills power to the drone. If the drone crashes, enter into emergency mode as soon as possible. This will avoid damaging the drone.
- Avoid water and other things that will harm the drone.
- Do not attempt to transport anything using the drone.
- You may not install/alter any software or hardware on the drone.
- The drones are fairly difficult to steer. Please do your best to avoid crashes. Always use the cover that covers all four rotors.
- You will be writing code to "maliciously" take control of a drone when someone else is piloting it. Make sure that this code also has the ability to immediately cut power to the drone if it gets too close to a living thing.
- If you are planning on doing something weird with the drone, ask the teaching staff first.

1.1 A Drone Controller {25 points}

Note that this portion of the homework is due on April 12th at 11:59pm.

Each group will be writing a controller to control the drones. The drone acts as a wireless access point (WAP). Your controller will operate on a group member's laptop (or, if you are really masochistic, a tablet or smartphone[1]). You should join the drone's WAP using the computer's built-in functionality. That is, your controller should assume that the computer is already connected to the drone's WAP.

Your controller should have the following functionality:

- The controller should enable the drone to take off and hover.
- The controller should enable the drone to land.

[1] The teaching staff doesn't have the resources to help you debug code that runs on a tablet or smartphone. If you choose this route, you're on your own.

- **The controller must have an emergency power-off mode.**
- The controller should enable the drone to adjust its roll (left/right tilt), pitch (front/back tilt), gaz (vertical speed), and yaw (angular speed).
- Your controller should be interactive – the user should be able to issue the above commands using single keystrokes (without having to press ENTER).

Here's the good news. A good portion of this part of the assignment (all requirements except for steering) are already done for you. You can base your code off of the teaching staff's controllers, which is available here.

Although you don't have to base your code off of the above, we strongly encourage you to do so. You do not need to cite our code. You may not use the Parrot SDK — your controller must craft its own control messages (see the teaching staff's controllers for examples).

And there's even more good news! There's a very good writeup of the drone's interface at: http://goo.gl/zZKDCn. You should read chapters A-D, carefully!

A tricky bit of the assignment is to take as input single keystrokes without requiring the user to press ENTER. Guess what? This is also done for you! See the above code for details.

Your controller does not need to have a GUI or show the streaming video from the drone. In effect, you'll be adding just the steering features to the provided code.

So, how can you build a controller without easy access to the drone? Fortunately, the fact that the drone uses UDP and that Wireshark "speaks" the AR.Parrot protocol makes this fairly easy. You can operate your controller without the drone and capture all messages using tcpdump. (Make sure to set the snaplength to be 0 to capture all of the payloads.) Then, using Wireshark, you can see whether Wireshark's interpretation of the message matches what you intended. Note that you'll need to use a recent version of Wireshark since older versions do not know how to parse AR.Parrot messages.

And, of course, you'll want to test out your code on an actual drone. You can do this during office hours, or your group can borrow a drone. There's more about the latter below.

All software will have to run on a group member's laptop or other computing device. You won't be using the *netid*-alice machines for this homework.

As with the other parts of this assignment, you'll need to both submit your source code and demo your software to a member of the teaching staff. The submission of the source code must occur before 11:59pm on the above due date. The demo can occur before or (shortly) after that time—you should either demo during office hours or schedule a time with a member of the teaching staff.

1.2 A Drone Subverter {35 points}

Note that this portion of the homework is due on April 21st at 11:59pm.

Here's where the fun begins.

For this part of the assignment, you will write a short program that causes a flying drone that is being operated by some other controller to land (not crash).

Why is this (somewhat) difficult? The drone can only talk to one controller at a time. Hence, your program will have to forge messages from the real controller in use.

Your program should work both against your homebrewed controller (see Section 1.1) as well as the regular controller App for Android or iPhone (see the respective smartphones' app stores[2]).

Your solution MUST forge packets that appear to be from the legitimate controller.[3] Since the OS normally writes the correct source IP address (i.e., yours, not the one you are trying to forge), you'll need to create raw sockets or use some library for packet injection (the library will abstract away the use of raw sockets). The difficult part is creating a valid Ethernet frame that includes a valid IP header followed by a valid UDP header followed by a UDP payload. This will be similar to the approach you used to build your port scanner detector, but in reverse (since you'll be sending these packets rather than receiving them). Once you have the packet formed, you'll need to inject it into the network. Raw sockets let you do this. You'll need admin privileges to create raw sockets.

You can use any packet injection library of your choosing. If you are sending raw packets from scratch, you'll probably want to read through a few online tutorials on raw packets, such as this one.

As with the previous part of the assignment, you'll need to both submit code by the due date, and schedule a demo with a member of the teaching staff.

1.3 Drone Wars {25 points}

Note that this portion of the homework is due on April 28th at 12:30pm.

It's time to race the drones. Your job is to get your drone from a starting position across a finish line. (The location of the race is TBD.) Your job is to cross that finish line before an opposing group gets their drone across their finish line.

You may launch network attacks against the other team's drones. Hence, you should also conjure up methods of protecting your drone from attack.

You may use any controller. For example, you can use the controller that you developed for the first part of this assignment, or the controller from the App Store, or something else. You will

[2] If no one in your group has a smartphone, you can test this during office hours.

[3] It is possible to "log into" the Linux instance that's running on the drone. Don't do this.

probably want to use separate devices for your controller and your attack machine.

This is fairly open ended. There are many ways to attack the other drone. However, you must attack it by sending network packets. No physical attacks are allowed. **Your attack should only affect the drone. It should not affect Georgetown's wireless networks.** You are also not allowed to log into the drone. Also, your attack should not be so severe that it "bricks" or otherwise makes the opposing drone permanently unusable. In other words, when we reboot the drone, it should work so long as no attack is taking place.

So how should/can you attack another team's drone? And how can you defend against attack? You'll need to figure this out as a group. Be creative. Here's a hint: study the drone protocol and search for potential weaknesses.

For the April 28th due date, you need to submit all the code that you'll be running on race day, as well as a document that explains your attacks and defenses.

The races will take place during class. The races will be structured as a single-elimination tournament. If neither team is able to cross the finish line during an individual race (i.e., because the opposing teams' attacks are so severe), we will re-run the race and disallow attacks.

This part of the homework is worth 25 points. 10 points are awarded for having an attack that works against an unprotected drone. 10 points are awarded for conjuring up some defense and properly documenting it. The writeup is worth 5 points.

Members of the winning team will each receive an *additional* 25 homework points. The second and third-place team members will each receive 15 and 10 additional points, respectively. You can think of these points as "extra credit" points.

Borrowing a Drone

Drones may be borrowed for up to 48 hours (during the work week) and over the weekend (due by 10am on Monday). Drones are borrowed by a "group", and a group may borrow only one drone at a time.

To borrow a drone, complete the form at http://goo.gl/forms/e62T6L6nJs. You must agree to certain safety and usage policies to borrow a drone. Note that you need not borrow a drone to complete the assignment; the drones are also available during office hours.

If you do complete the form, please then send a request, via Piazza, to the instructors. We will arrange a time during which you can pick up the drones.

We emphasize that a borrowed drone may only be operated indoors, and only by members of the group, and only under safe conditions. Intentional violations of these policies will result in immediate confiscation of the drone and a 0 on this homework assignment. Additional sanctions are also possible.

Submission Instructions

To upload the various parts of your assignment, navigate to the COSC235 course on Blackboard, click the “Assignments” link on the left hand side, and select the appropriately labeled option (i.e., “hw3-drone-controller”, “hw3-subverter”, and “hw3-dronewars”). For each part of the assignment, include source code (if appropriate) and writeups/documentation (if appropriate) in a single compressed tarball (.tar.gz file). When submitting source code, include a description of the platform on which your code is designed to operate (e.g., Linux, Mac with certain libraries installed, etc.). Note that all parts of the drone assignment must be demoed either during race day or during office hours (or special arrangement).

All writeups and documentation should be submitted either as a PDF or as plain (not RTF) ASCII.

Only one group member per group needs to upload files to Blackboard. As described above, all group members will receive the same grade.

Please post questions (especially requests for clarification) about this homework to Piazza.