1 What Did We Learn?

1.1 Dominic

As far as technical information is concerned, I (re)learned many things about ROS and Linux, in particular creating and building ROS packages along with ROS process control and the publisher/subscriber system. However, I feel that the real learning for me was in non-technical areas, particularly communication and scheduling. First and foremost this project reinforced the idea that if Im producing work for someone, they should see it before the deadline, so they can tell me what I missed ahead of time instead of after. This was less of an issue for me personally then for the team as a whole, especially with our documentation, but the point stands. Our whitepaper was not up to snuff, nor our Spring midterm documentation, nor our design document and tech review, not even the blog posts apparently, and all of that could have been avoided if someone had looked it over before submission and told us we were completely off base. However, none of this happened because we tended to cut real close to the deadline on all of these documents, hence scheduling: starting in earnest ASAP instead of half-heartedly starting it and only really getting to it with 2 days to spare would have made it much easier to present a full draft to those in charge before the deadline instead of just getting it submitted with hours to spare. Much of the delay for those papers had to do with factors outside of my control, which was unfortunate, but the point remains that better planning could have helped with this. The biggest thing I learned about project management is that if you arent managing the project, its best to assume nobody is until evidence is provided otherwise. Most of spring term felt like I was attempting to get everyone involved on the same page trying insure the team knew what was expected of us and occasionally moonlighting as a ghostwriter for Emily. On that note, I learned a lot about project management, or at least the interpersonal communication part, not so much the planning part as by spring term it was too late to plan. Ironically, this played out like it normally does for group work for me: Im in charge of confirming requirements and relaying information between my group and those on high, and as long as I get my share of the work done things normally play out alright from there. If I got a redo, Id first and foremost redirect this project away from research, as I still cant justify this as a research paper. I would probably just make sure Vee nipped the threat model thing in the bud before it became the purpose of the project, as neither of us expected it to work out this way. Next, I would have never bothered with the drone, as I doubt it would have been useful even if we had gotten it to work and it took way too much of Zachs time. Last, I would have tried to look for a heartbleed-esqe bug in the ROS subscriber-publisher system for arbitrary code execution, because I think thats the real holy grail as far as OS based attacks. Everything I did relied on already having the ability to execute arbitrary code on the ROS system, which is a rather large assumption to make.

1.2 Emily

This project taught me more than I ever thought it would in just about every way. On the technical side I learned first and foremost about the core of our project, ROS. I had absolutely no robotics experience going into this so it was a fantastic opportunity to learn about a whole new field and study it in depth. I also hadn't had much experience with drones before this, so it was really cool to be able to play with some high quality hardware and learn how it worked. I did have security knowledge coming in, and applied it vastly in the creation of our exploits, but in writing each one of them I gained quite a bit of new knowledge about the portion of the system with which I was working. Being able to gain all of this technical knowledge was amazing, but I think the most important learning was in the more non-technical realm.

I've always loved research and wanted to do it on a large scale, but I truly underestimated how difficult it is to do on a project longer than a term. I really had to learn (by failing time and time again) how to best manage our time and expectations. Naively I thought we would be able to accomplish a lot more, and do it much more smoothly and quickly. One of the most difficult lessons of this whole project for me was learning how to ensure that I was serving the expectations of everyone and their requirements equally, and not just focusing on what I considered to be the most important. It's difficult for most people to get serious work done on things that they don't think matter as much as something else, and I learned a great deal about trying to balance that.

On smaller scales I've often ended up managing group projects so I somewhat fell into that role in this, but it was a lot harder to juggle with this one. Time management was challenging, and keeping track of everything at once made this even more difficult, but after each term I felt I had gotten better at it. It was also clear that knowing communication is crucial and actually communicating effectively are two different things, with it being much easier said than done. If I had to say I learned the most about anything from this project, it was about communication in project management and how difficult it can be when not done perfectly.

If I had to do it all again, I would have everyone meeting much more regularly, and working together as a group in the same room. Of course having varied and busy schedules made this tough, but we could have done better.

I would also have started just about everything sooner. Humans are notoriously bad at time estimation, and this project truly proved that to me. I'm still happy with the experience I had though, because it taught me so much, from technical details to collaboration. While I may not have loved it at many points in time, I greatly appreciate my capstone experience in retrospect because I needed the challenges it provided.

1.3 Zach

I learned a lot working through this project this year, through all the ups and downs, I walked away knowing more than I did when I started. In terms of technical information, I learned the importance of a threat model in the information security field and useful it is when guiding how to approach a security evaluation of a system. I also learned a lot about drone hardware, flight controllers, the Beagle Bone Black, and of course, the Robotic Operating System. Along with ROS, I learned how to develop ROS packages, and feel much more comfortable developing for the platform. I also learned about information security methods and tools, such a fuzzers, and the role they play in trying to find vulnerabilities in a system.

In terms of what I learned on the non-technical side of things, I would say it mostly had to with how to conduct a research project of this scale. Before this project, I had never been part of an actual research project before, so having this exposer really taught me the importance of project management and communication.

With regards to project work, I learned the importance of planning, time management, team work, conflict management, and the importance of communication amongst all members and entities involved with the task at hand. Planning what needs to be done, and sticking to a productive schedule helps to keep the project on track; I did not realize how important these two things were until I found myself wishing I could go back and plan things better. It also became apparent that communication is key, and it is important to take the extra step to make sure that everyone is on the same page regarding what is going on in the project; there were many issues that came from lack of communication from all parties involved with this project, and I would say that for me, this was the biggest lesson I learned throughout all of this. Always communicate, and communicate often. With these issues, I learned how to deal with conflict and how to better listen to the input and needs of my teammates; this was the first time I really faced project conflict, so it was another important lesson during my capstone experience.

If I could go back and do it all over, I would spend much more time and care with the initial research and planning phase of the project. I feel that a lot of the problems that we ran into could have been avoided if we focused on more in-depth research and planning, especially regarding our hardware, instead of being surprised when nothing worked as expected. I would also take more care with making sure I understand what the client I asking of me and the team, and would make sure that the client and our team is committed to sticking to the goals we set during the planning stages. I would also ask for assistance and clarification much sooner, instead of continuing to try to fix things myself; not only did this waste a lot of time, but it hindered the project as a whole. Lots of lessons were learned during this experience, and I feel like I am walking away more prepared for my future.