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-esk 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.