Reflective piece

In this piece, I’ll write in chronological order about my reflections and experience during the Secure Software development module.

This module revolves around a group project, divided into two parts, first is to be designed a document representing an application chosen among three cases. The following part is the development of the said application. In the first units, we get to discuss UML, which was my duty during the design phase. I learned a lot during these combined experiences, partly because I enjoyed exploring UML and partly because I got to focus particularly on that. The communicative power of those models is amazing, they can be read and understood by non-technical actors as well as clarify to technicals about requirements and structure. Talking about Codio activities we went through tools that improve your python coding in terms of styling and bug prevention, I got to discover these as they are much more important than one could think, readability and maintainability of code is a big aspect in large code bases. The buffer overflow exercise was useful too to dig into memory allocation and its security implications, fortunately, Python manages memory by itself which is more secure for developers, although limiting in certain scenarios. We get to the end of the first half when we need to submit our design document. As said before this was an amazing learning opportunity for me, the team worked well together and I’m satisfied with both my contribution and my team.

The following weeks revolve around operating systems, APIs and the development part with a bit of cryptography. I particularly enjoyed this part. Starting from cryptography, I enjoy this topic, exploring the different ingenious techniques through the suggested reading and experimenting with the Python libraries which are some amazing work, I did the cryptography part for our project with an amazing library called bcrypt that I used in JS before. There was a discussion about the TrueCrypt case which I found interesting and was integrated well with the operating system part, I got the chance to learn a bit about how to protect your personal data inside your own machine. APIs are a topic of interest for me as I am working as a junior web developer. Although Python is not the preferred language for web development, it is gaining more and more power in that field, and with this module, I got to explore how good it is becoming and yet another way Python can be used. Flask is a great library that replaces ExpressJS and its similars, I don’t think it is better yet but is a completely viable option. There is still so much to learn about Python handles different types of databases such as NoSQL and that would be interesting. From the precedent module, where I learned how well Python behaves with OOP (I always used it in functional programming), to this I had the chance to explore its versatility and get another point of view on these topics provided by the Pythonic way of doing things which compared to JavaScript is much easier at times. Toward the end, there was this discussion about microkernels and monolithic kernels. Without exaggerating I loved the format of this discussion. The suggested reading about the emails exchanged between Andrew Tannenbaum and the creator of Linux, Linus, was as informative as it was fun to read. I think that a spark of humour and irony can make everything more enjoyable, especially when learning on the right occasions. That discussion touched such interesting topics in such an enjoyable way that it was a great reading suggestion, the best so far. Just like the majority of people probably I have a hate-and-love relationship with windows and I reckon that macOS is faster and lighter, but both systems rely on hybrid kernels, with macOS being based on Unix and the Apple-produced kernel, I enjoyed reading about the ancestor of today’s neverending discussion about Windows versus macOS and Linux.

In conclusion, I want to talk about the group project. This was a fun and instructive experience with some downsides. I think this is one of the best ways to learn both programming and team dynamics. The division of duties, getting to experience more team dynamics and leveraging such incredible tools for teamwork (Git, GitHub and more) give learning a great boost. We started a bit slow but picked up the pace in the end. I got to mention that I loved some of the pair programming sessions we had, although very time-consuming, they were effective in getting things done. We did not manage to include everything we wanted because we were blocked for an extended period by some unfortunate inconveniences that I won’t mention here as I already did in the preposted place. In the end, I don’t feel to condemn anyone for the missing features of the ending results because we gave our best for this project and everyone that contributed used so much of their limited free time that I still see the ending result, considered the limited human resources we had (three members developed this project instead of five), as a good result.

To summarise this was the most time-consuming module yet, but the most instructive and interesting one. It was an amazing learning opportunity and I took as much out of it as I could with my time, I certainly did miss out on some things but I am satisfied with how it turned out and the knowledge it gave me.