Narrative

C++ Secure Coding

Caroline Wilson

CS-499

June 2021

I chose to polish a group of three .cpp’s that represent my abilities in the C++ language, as well as showing my ability to build, utilize and call containers in the creation of a database for an auction. Through this I was able to compile a data base and structure it so that it was in order by bid amount, then using a hash table and a user input program search, add, remove, and sort the bids that are in the container. This project was initially created in February 2020.

I found that this artifact was important to my portfolio because of the necessity for the ability to create and maintain containers in the field. I was told that this would be a welcomed addition to my skillset as I move forward with my career in cyber security. These programs show my ability to structure proper containers and the ability to maintain them with the C++ language. This language, while basic in comparison to some, gives an easy jumping off point to other languages, such as python, and has an easy meld with data languages such as SQL. By beginning the structure of these programs, I would be able to move seamlessly into similar programs that showcase my abilities in other languages.

. As far as this adaptation, the biggest overhaul was that of continuity in structure and notation. There was disparity between the three projects as well as differences line by line within the original code. Readability and a basic understanding of the code to allow for modularity is imperative to the successful completion of any project. It was important to me that I reflect my attention to detail and best practices. With the code not being representative of my attention to detail, I had to rife through it to ensure the proper notations were in place, and that it wasn’t getting convoluted with unnecessary bloat.

The biggest take away for me as I was reflecting on this project was how far I have come over the past year in my developmental challenges. When I was starting on this project, a year and half ago, so much of the syntax and command lines were Greek to me. Though I was able to create a quality product, I found that I was stumbling over my own reasoning and application of the commands that I was trying to learn. At that point, when I would read back in my code to see where I was, to test it (to a very minimal scale), and to ensure that it would run as I intended, I would find myself lost in the order and the syntax of the command and my eyes would glaze over at times trying to understand where I was in the code and the reason for the choices in algorithms I had used. However, now when I read over the lines of code, I find that I understand it much better than I had at that point. This was surprising to me, even though I had taken this class last February, I almost had a better understanding of the material now than when it was fresh in my mind. In addition to the basest of understanding, I saw potential and inspiration to prove the modularity of some other projects in the testing and deployment potential of these programs. I see a way in which I can construct my portfolio in a linear fashion in order to best reflect my strengths.