Joshua Battaglia

Professor Goggin

CS-499: Computer Science Capstone

20OCT2024

**Self Assessment**

**Part-1A:** Completing my coursework throughout the computer science program and while developing this ePortfolio I was able to shape my professional goals and values. I learned that I enjoy finding solutions to common problems and using technology to make repetitive tasks easier such as creating scripts for my Linux distribution at home and in the workplace. Throughout this portfolio, I enhanced a single program from start to finish to showcase how I can work from beginning to end on a single project. I took inspiration from a script I created, located here: <https://github.com/batspp/AutoUpdate-Upgrade>which was written in a shell language and can be used to create a custom command by entering a symbolic link in the “/usr/bin/” directory. This simple script and instructions eliminate the repetitive steps needed to update and upgrade a Linux distribution.

            As a Sr. Systems Engineer operating within an Agile framework, I know time is important, and delivering value to the customer sooner helps establish a solid collaborative environment. By automating and scripting repetitive steps you shorten the amount of time required to deliver a usable product by simplifying the process with a script you eliminate time spent repetitively typing, and the possibility of human error. The simplification of these types of processes inspired me to take it a step further by limiting the possibility of a user error and presenting simple layouts that are intuitively understandable for an average user.

            I once again took this further and ensured that the technical side of my project was also simple and reusable by building custom packages to implement the application’s frontend and backend functionality. Now the main Python file only requires one function imported from two packages (gui, and db). While working through the application I worked incrementally trying to think of what type of tool or package would make my experience better when attempting to add new functionality.

**Part-2A:** When collaborating in team environments one of the most significant obstacles is communication. I currently collaborate with other contracts at my current site, and even being in the same building doesn’t make communication any easier. Documentation is a great way to enhance communication by providing clear and concise descriptions for almost any aspect of a project. A good example of using documentation is how I use a software description document (SVD) to install software. When any SVD is neglected to be updated there is such a large impact since effort now has to shift focus to figuring out why something isn’t working properly which wastes time and man-hours.

            When communicating with stakeholders we can often build on the documentation we previously developed and tailor it to a new audience. Being able to explain and translate technical documentation to non-technical stakeholders is fundamental to maintaining a collaborative environment with the customers. This type of communication also greatly contributes to delivering value to the stakeholder sooner since this allows them to provide a status report to their management.

            While developing software it is important to address many different components such as data structures and algorithms, databases, and security. One of the most important aspects is security and it should not be forgotten about. Data structures and algorithms need to be properly optimized since security will create a slight trade-off for the software’s development. Extra steps should be made so that adversaries cannot exploit the software architecture such as with buffer overflow and underflow, and SQL injection that can allow a user to gain unauthorized access to sensitive data. We also need to ensure that the application is still usable so defense in depth is a great way to secure any technology that uses sensitive information such as banking software.

            The same security mindset applies to databases since you do not want any sensitive information stored in plain text within a database. What’s great about using a database is that sensitive information doesn’t have to be stored in memory. An experienced threat actor could exploit memory and find any cookie crumb of sensitive information within memory even if it’s stored for a short amount of time.

**Summary**

In this portfolio, I enhanced a single artifact based on a legacy binary application that I reverse-engineered for a previous artifact. Throughout these enhancements I:

-Employed strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science by completing the following enhancements **(link)** by using docstrings throughout my custom packages as documentation and communication. I can inform non-technical audiences how the package works, and how it is used throughout the application. Each function also contains a docstring for any developers looking to enhance or customize their version of the application utilizing the packages.

- Designed, Developed, and Delivered professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts by completing the following enhancements **(link)** with code commenting, and a video code review describing the status of the current artifact and planned enhancements. I also ensure naming conventions make sense throughout the enhanced artifact such as gui, db, and auth.

-Designed and Evaluated computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices by completing the following enhancements **(link)** by improving the time complexity of the artifact, and the main function.

-Demonstrated an ability to use well-founded and innovative techniques, skills, and tools in computing practices to implement computer solutions that deliver value and accomplish industry-specific goals by completing the following enhancements**(link)** by porting the artifact to Python, creating custom package, and used existing modules to implement database operations.

-Developed a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources by completing the following enhancements **(link)** parameterizing database operations to prevent SQL injection, and porting into Python to take advantage of the programming languages automatic memory management. I also ensured sensitive information such as the user password is not stored in memory at any given time and is properly salted and hashed before being stored in the database.