Joshua Battaglia

Professor Goggin

CS-499: Computer Science Capstone

19OCT2024

**Data Structures and Algorithms Narrative**

            The artifact I chose is a login portion of a console application derived from a legacy investment application. The original binary was disassembled, analyzed, and converted into a C++ console application. For the legacy application, a user had to log in using a username and password to validate credentials and gain access permission to some investment data. This artifact was created on 23 August 2024 to demonstrate my ability to reverse engineer legacy applications into a modern coding language.

I chose to use this artifact because I discovered a lot of security vulnerabilities while analyzing the assembly code of the binary file. Securing these vulnerabilities for the C++ conversion added complexity to the original application and increased time complexity to be a trade-off since user input had to be validated (to prevent possible exploits). A loop continuously evaluating user input caused the time complexity to be O(n) at its best case and O(n\*m) at its worst. Porting this login application into a Python application that uses Tkinter (a C library for graphics) is a great demonstration of my ability to understand a software application's overarching use cases while implementing them into a user interface (UI). Implementing a UI for the application makes it easier to understand and appears more professional than a command-line application.

During this enhancement I demonstrated my ability to design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solutions while managing the trade-offs involved in design choices by improving the time complexity of user validation processes within the application. Secondly, I employed strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science by utilizing docstrings throughout my Python files which can be used as documentation for the application. Including technical information describing each function with docstrings allows other developers to understand and enhance the existing code in the future without having to directly interface with the originator.

While working through this enhancement process I learned a few tools and concepts. While porting to Python and using existing libraries, I learned about Python's “help()” function. Although unfamiliar with Tkinter (a C-based Python library), I was able to use the help function within a Pycharm terminal to find resources and functions to use during development. I also learned more about standard Python practices, such as namespace checking and utilizing “\_\_init\_\_.py” files to limit access to the custom package code. The biggest challenge I faced was using a new tool in a way that wasn’t explicitly documented within that library’s resource documentation. My personal goal with this project is to maintain simplicity for both technical and non-technical aspects of the application, which is a challenging balance.

