Portfolio Summary

Arizona State University
Christopher Richard Bilger
ASU ID: **REDACTED**

Abstract - This paper will compile a summary and two portfolio reports into a single, final portfolio submission for consideration of completion of the Master of Computer Science (Cybersecurity) program.

1. Summary

The following two sections will highlight and explain the individual portfolio reports that have been written for the consideration of completion and approval for the Master of Computer Science (Cybersecurity) program and degree, respectively.

2. Portfolio Report #1

The course that is being used for the first portfolio report submission is *CSE 511: Data Processing at Scale*.

The projects for this portfolio report made use of modern toolings, such as the Scala programming language, the Spark big-data analysis tool, IntelliJ, and the Python programming language.

Problem:
Using the Scala programming language, alongside the Spark big-data analysis tool, my team had to solve two tasks. The first was to compute the boundaries of a 2-dimensional rectangle, and the second was to join the code from the first task with a functional SQL database querying system.

Solution:
For the first task, my team wrote functional code to compute the boundaries of a 2-dimensional rectangle as well as determine if the Euclidean distance between a set of points lay within a range. The second task's solution was to combine the code from the first task alongside geospatial coordinates to determine the top 50 coordinate locations near specified coordinate locations.

Knowledge Gained:
Throughout my time working on these projects I learned new coordinate and geospatial referring systems, and I also learned a new programming language (Scala), as well as how to use Getis-Ord scoring and ranking.

3. Portfolio Report #2

The course that is being used for the first portfolio report submission is *CSE 545: Software Security*.

The project for this portfolio report made use of modern containerization methods, namely Docker, for running and maintaining the Capture The Flag services within each group's virtual server.

Problem:
Given a virtual server that is running numerous networked services, find and patch potential vulnerabilities within those services while, at the same time, exploiting those same vulnerabilities on the opponent's virtual servers.

Solution:
My team created a script runner program that automatically executed user-created Python 2/3 scripts to deploy exploits against opponent virtual servers. This automated script runner proved extremely valuable as the event occurred over a continuous 24-hour period. Throughout the event, my team would patch our services while simultaneously creating exploits against opponent services. We would then add those exploit scripts into the script runner to continue running until the event came to an end.

Knowledge Gained:
By far the number one skill that I gained throughout this project was that of communication being vital to the success of an operation. My team maintained communication throughout the entirety of the 24-hour event.