# Portfolio Report

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 event.

# CSE 511 - Data Processing at Scale - Portfolio Report

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

*Abstract - This paper will review the problems, solutions, design considerations, personal contributions, and lessons learned from the project given during the CSE 511 course.*

## 1. Introduction

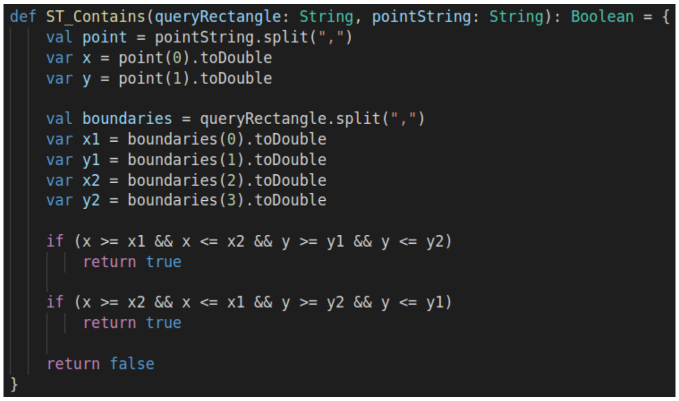
The projects for CSE 511 Data Processing at Scale started with small, introductory programs that were used to introduce the team members to one another, and also to introduce each team member to the development environment setup, the Scala programming language[3], the Spark big-data analysis tool[4], IntelliJ, and the Python programming language. Through this portfolio report, I will cover the projects completed in this course, my role in completing those projects, and what I have since learned and continue to use from these projects.

## 2. Solutions

When my team set out to work on this project, we decided to solve phase 1 and phase 2 individually before meeting and discussing our solutions. This allows me to easily show what I did to solve the problems that I encountered in each phase of the project. I will split up my solutions into two sections, one for each phase of the project, and explain in detail my solutions to the problems given.

**Phase 1:**

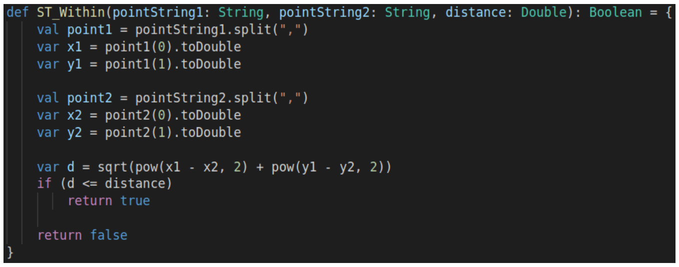
For the first phase of the project, I had to create two functions. One of these functions had to return a boolean value of whether or not an **x-y** point resides inside of the boundaries of a set of **2 x-y** points, creating a rectangle. The first step that I did was to convert the single-point string into an **x** variable and a **y** variable. I then converted the longer multi-point string into two separate **x-y** variables, consisting of the corner points of the rectangle. Once the conversions were finished, I checked if both the **x** and the **y** variables resided inside of the **x-y** bounding box, and returned a boolean representation of this value. Below is an image of the steps, in order, outlined above.



```
def ST_Contains(queryRectangle: String, pointString: String): Boolean = {
    val point = pointString.split(",")
    var x = point(0).toDouble
    var y = point(1).toDouble

    val boundaries = queryRectangle.split(",")
    var x1 = boundaries(0).toDouble
    var y1 = boundaries(1).toDouble
    var x2 = boundaries(2).toDouble
    var y2 = boundaries(3).toDouble

    if (x >= x1 && x <= x2 && y >= y1 && y <= y2)
        return true

    if (x >= x2 && x <= x1 && y >= y2 && y <= y1)
        return true

    return false
}
```

Fig. 1. Phase 1, function 1 code implementation

The second function had to return a boolean value of whether or not the Euclidean distance[2] between two points was less than or equal to a given distance **d**. The first step that I completed to solve this problem was to convert the two point-strings into **x** and **y** variables, respectively. I then calculated the Euclidean distance between these two points, before checking if this distance is less than or equal to the given distance **d**. I returned the boolean value of this logical comparison. Below is an image of the above steps, in order.



```
def ST_Within(pointString1: String, pointString2: String, distance: Double): Boolean = {
    val point1 = pointString1.split(",")
    var x1 = point1(0).toDouble
    var y1 = point1(1).toDouble

    val point2 = pointString2.split(",")
    var x2 = point2(0).toDouble
    var y2 = point2(1).toDouble

    var d = sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2))
    if (d <= distance)
        return true

    return false
}
```

Fig. 2. Phase 1, function 2 code implementation

**Phase 2:**

The second phase of the project consisted of joining the contents of the first phase along with SQL database querying as well as some mathematical analysis to find the solution to the problem given. Below I will outline the steps that I took to solve this problem.

**Entrance.scala:**
I added our group number to the "appName" function-call parameters.

**HotzoneUtils.scala:**
I copied over the "ST_Contains" function that I wrote in phase 1 of this project, into the "HotzoneUtils" object.

**HotzoneAnalysis.scala:**
I realized that the "runHotZoneAnalysis" function definition was only missing the correct return value of the hot zone data frame. I took the "join result" data frame, grouped by the "rectangle" keyword string, and then iterated over this data frame to find and sort only by the "rectangle" keyword string.

**HotcellUtils.scala:**
I added a calculation function to find the number of adjacent hot cells for a given **x-y-z** coordinate position. To solve this, I had a simple counter mechanism that started at an initial value of 0 and would then increment by 1 if and only if the given **x-y-z** coordinates lie on either the minimum **x-y-z** value or maximum **x-y-z** value. I then iterated through the possible counter values and returned the corresponding integer value.

I also added a function that calculates the Getis-Ord score, or the amount of point clustering that is occurring, in a given area. This function takes in numerous parameters; such as the total number of cells, total number of hot cells, x-y-z coordinates, etc. and outputs a double value corresponding to the value given from the following formula[1]:

$$G_i^* = \frac{\sum_{j=1}^{n} w_{i,j} x_j - \bar{X} \sum_{j=1}^{n} w_{i,j}}{S \sqrt{\frac{\left[n \sum_{j=1}^{n} w_{i,j}^2 - \left(\sum_{j=1}^{n} w_{i,j}\right)^2\right]}{n-1}}}$$

Fig. 3. Getis-Ord formula

$$\bar{X} = \frac{\sum_{j=1}^{n} x_j}{n}$$

$$S = \sqrt{\frac{\sum_{j=1}^{n} x_j^2}{n} - (\bar{X})^2}$$

Fig. 4. Two sub-components of the formula are in Fig. 3.

**HotcellAnalysis.scala:**
After writing one too many solutions to this problem, which I see as the biggest problem to be solved for this project, and still having errors, I re-designed my solution from the ground up. The function "runHotcellAnalysis" was missing the database querying functionality so that is was I decided to add first. If anything, I would be able to solve that problem and, at the very least, obtained data from the database which can then be converted into the information that I am looking for. That is, the top 50 pickup coordinates are aggregated and sorted by their Getis-Ord score. Once I solved the problem of being able to connect to and query the database, I calculated the Getis-Ord score from the previously obtained database query results. I solved my initial problem by re-thinking the function as a database query which I could then convert and calculate the necessary information from. I, like many others, had another problem that needed to be tackled. When I ordered by descending Getis-Ord score I was still getting incorrect results on the Coursera AutoGrader. I noticed that this was correctable by explicitly ordering all of the results by

not only descending the Getis-Ord score, but also by their x, y, and z coordinates.

## 3. Results

Throughout the time that I spent working on the projects that are covered in this portfolio report, I came across numerous findings that I would classify as intriguing and useful for would-be students that might also take this course. I will split them up into two sections; the first section is my findings from the first half of the project, which is up to and includes the assignment labeled “Project Milestone 4”, and the second section will cover the portions of the project after “Project Milestone 4”.

**Section 1:**
I found the project discussions, especially the discussion on SQL versus NoSQL database, to be incredibly insightful concerning big-data processing and operations on big-data as the scale of the data drastically increases. For the “Project Milestone 4” assignment, which was completed by each team member individually before our meeting and discussion of results, I found that this project had large-encompassing use-cases in the real world. I tend to look at how a project can directly relate to both my studies as well as my future career development. If others go into this project with that same sense of open-mindedness and overall interest in the subject matter being taught (Data Processing at Scale), then I think that this project will help to show to them how many uses spatial queries have in both the real world and the theoretical world.

**Section 2:**
This section covers one portion of the project and that is “Project Milestone 5”. At face value, this part of the project appears quite daunting, but I think that for newcomers to the Scala and Spark world, this is an excellent test of one’s ability to pivot and learn something new. I found the Hot Zone Analysis very interesting. I learned that it is possible to calculate the relative scale, or “hotness” in this case, of individual 2-dimensional rectangles using Spark and a small amount of common SQL. Again, as I stated above in section 1, I found that when I went into this part of the project with an open mind, I was able to learn more about the underlying structure of the data and how it can directly influence everyday applications.

## 4. Contributions

I think that my impact on the overall group project was fairly substantial. As explained above, each team member worked on each assignment individually until we each had a solution to the problem at hand. We then met up and created Zoom conference calls to share our findings and then to submit the group portions of the project. My contributions span each portion of the group project. Below is a detailed listing of my contributions towards the successful and timely completion of each project assignment.

**Project Milestone 1:**
This is a shorter section of the group project; however, I aided in summarizing my thoughts, as well as the group discussion, into the submitted file. I joined the group discussion open-minded. I shared my insights and learned from my team members through their anecdotes.

**Project Milestone 4:**
I positively influenced the source code that my team ended up submitting for the assignment and taught my team members what I had learned about the Scala programming language while completing this assignment on my own.

**Project Milestone 5 - Source Code:**
I wrote approximately half of the “README.pdf” file that was submitted along with the source code for this assignment. I also wrote approximately half of the source code, tested the final group submission source code, and compiled the final “zip” file that was submitted for this portion of the group project.

**Project Milestone 5 - Systems Documentation Report:**
For this portion of the group project, each team member took one section of the report, wrote that section, and then we compiled the individual sections before submitting the final draft. My portion of this assignment to write was the “Design Solution & Methodology - Phase 1” section. Along with writing this section of the Systems Documentation Report, I also drafted the overall structure of the report as well as finalized the contents of the report.

### 5. Lessons Learned

My work on this project greatly contributed to my computer science and software engineering knowledge and skills. I learned an entirely new programming language, Scala, which I had heard of in the past but did not expect nor anticipate ever learning. I find this very rewarding because I want to continually learn and evolve my skill sets in the field of computer science. My day-to-day work is in the web development space, so using the full-fledged IDE IntelliJ was a newly acquired skill that I will immediately transfer into my career. Before taking this course, I had very limited knowledge of large-scale data and the processing of large-scale data. Now that this course is wrapping up, however, I can proudly say that my subject matter knowledge of data processing, large-scale operations, SQL, NoSQL, etc. is greatly improved and I am very confident that I will be able to use this knowledge going forward in both my academic as well as my professional careers.

**Team #7:**
Christopher Bilger
Benjamin Parrish
Balaji Radhakrishnan
Ashraf Sayyad
Chirag Sindhwani
Jebaraj Vasudevan

## References

[1] "ACM SIGSPATIAL Cup 2016." ACM SIGSPATIAL GIS Cup 2016, ACM SIGSPACIAL, sigspatial2016.sigspatial.org/giscup2016/problem.

[2] "Euclidean Distance." ROSALIND, rosalind.info/glossary/euclidean-distance/.

[3] "The Scala Programming Language." Scala-Lang, www.scala-lang.org/.

[4] Apache Spark™ - Unified Analytics Engine for Big Data, spark.apache.org/.

# CSE 545 - Software Security - Portfolio Report

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

*Abstract - This paper will review the problems, solutions, design considerations, personal contributions, and lessons learned from the project given during the CSE 545 course.*

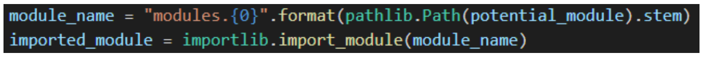
### 1. Introduction

The project for CSE 545 Software Security was to design and implement tools that would aid my team (Team 31) during the Project Capture The Flag event. During my team's preliminary Zoom conference calls, I had the idea of creating a script runner in Python. The intent was for the script runner to automate and drastically simplify the process of discovering vulnerabilities in the running services, writing code that will exploit those vulnerabilities, and creating patches for our team's services to remove the vulnerabilities that we found. Through this portfolio report, I will cover the project completed in this course, my role in completing the project, and what I have since learned and will continue to use from the project.

### 2. Solutions

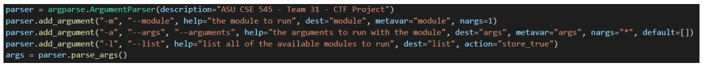
The end product that my team ended up creating was a modular (by design) Python 2/3 script that enabled us to create and run various other scripts and tools. Each of the individual scripts/tools that we created were able to be used stand-alone or through the script runner. Below, I will explain the design and implementation of the script runner as well as some of the individual modules. There are numerous other scripts/modules that we created for this project, but they ended up not being as useful during the CTF event as we would have preferred.

1. For the first section of this project (the script runner[2]), we worked on implementing a modular import system that would import all python files in a specific direction and then would run those python scripts with the default and user-supplied arguments. During this section of the project, we also created a template script that aided in simplifying the future creation of scripts/modules. The first two images below are showing how the modules are dynamically imported and how the script runner argument system functions. The argument system that we created takes in an argument **m** as the module name and can handle optional arguments, **a**, in the form of [arg1] [arg2 [...]. There is also an optional list argument that will display helpful information about all of the dynamically imported modules.



```
module_name = "modules.{0}".format(pathlib.Path(potential_module).stem)
imported_module = importlib.import_module(module_name)
```

Fig. 1. Code snippet showing the dynamic import of script-runner modules

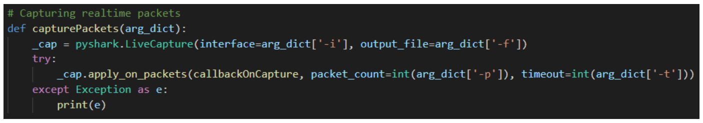


```
parser = argparse.ArgumentParser(description="ASU CSE 545 - Team 31 - CTF Project")
parser.add_argument("-m", "--module", help="the module to run", dest="module", metavar="module", nargs=1)
parser.add_argument("-a", "--args", "--arguments", help="the arguments to run with the module", dest="args", metavar="args", nargs="*", default=[])
parser.add_argument("-l", "--list", help="list all of the available modules to run", dest="list", action="store_true")
args = parser.parse_args()
```

Fig. 2. Code snippet which lists the available command-line arguments for the script-runner

2. The second section of the project was the longer of the two sections. This consisted of designing and programming the individual modules that we intended to use during the CTF event. Below I will explain the solutions that my team came up with for the net-analyzer and the template-exploit.

3. The “net-analyzer”[3] script was intended to be a network traffic analyzer that we would use to check whether specific network activity was from the game bot or the other teams. This module unfortunately was not very helpful during the CTF event because we didn’t end up needing to analyze any network traffic or activity. I think that this module would be more helpful during a CTF event that has a higher emphasis on network-related detection and authorization. If we had more time during the CTF to focus our efforts on network-related data processing, then I think that we might have been able to detect and identify the packets from the game bot to separate them from the other teams. The idea was, that we would then be able to block packets from other teams while still enabling the game bot to operate as it normally would. Below is a photograph of the “pyshark” Python module capturing network packets in real-time and sending them to a handler function. As you can see from the picture, we also allowed the user that is running the module a high degree of customizability for the packets that they wanted to capture.



```
# Capturing realtime packets
def capturePackets(arg_dict):
    _cap = pyshark.LiveCapture(interface=arg_dict['-i'], output_file=arg_dict['-f'])
    try:
        _cap.apply_on_packets(callbackOnCapture, packet_count=int(arg_dict['-p']), timeout=int(arg_dict['-t']))
    except Exception as e:
        print(e)
```

Fig. 3. Function showing how packets are captured in realtime with configuration capabilities

4. The “template-exploit” script was extremely helpful during the CTF event. This script automated the execution of the exploit that we supplied to it as well as automated the submission of flags to the SWPAG[4] endpoint. This script was designed from the beginning to allow the exploit creator to easily copy and paste the template to a new python file, add their exploit code, and then run the script to

see if it was a successful exploit. Below is a picture of the template opening a connection to the SWPAG client, iterating over all of the teams' services, running the exploit code for our team to obtain the open ports, and information about those ports that the individual services were running on. We were not aware until it was already designed and built, that the service ports would be known in advance and that running port scanning was essentially unnecessary for this competition. I think that this script is still very useful for other competitions where the service ports are not known in advance because it will allow the team(s) running it to find port information very quickly and without the other teams knowing that we are scanning them.

## 3. Results

I had some very interesting findings occur during the creation of my team's project as well as the usage of our project during the CTF event. First of all, the project that we made was highly modular and horizontally scalable, meaning that adding and removing modules is incredibly easy to do. This lent itself to efficient usage of our time during the CTF event because our tools became an extension that we could use, but that didn't distract us from our tasks. I found that creating tools before the CTF event, and I would imagine all CTF events, allows the players to communicate much more effectively and to work on vulnerability analysis and patching those vulnerabilities. I think that this project and others like it are incredibly important as they allow you to "dive straight into" the CTF competition without fumbling around with trying to manually backup services, install programs and plugins, get your environment setup, etc. Once the CTF event was over I read through our codebase and found numerous places where we could have lowered the tool's friction even more. I found that if I were to go back and redo my team's project with the knowledge that I have now I would have spent time exploring open source tools that have already been written for CTF competitions and used those as a baseline for the specific type of tools that are used most frequently. One final note about the results that I found from my team's project during the CTF event is that our project guided us to find the vulnerabilities of our services and then to exploit them on the other teams. That is, our project gave us a tremendous amount of flexibility for open communication and teamwork as opposed to guiding us to work independently, as I think some other projects might have been doing.

## 4. Contributions

I think that my overall contributions to the group project aided greatly towards our CTF event success. Through my time spent working on this group project, I contributed to numerous individual submissions. I think that I played a huge role in the development of our team's vision for our project as well as the implementation of both documentation and code. Below I will explain in further detail the individual contributions that I made towards the success of my team's project.

**Team Project Proposal:** I wrote up the majority of the team project proposal document, which I then later submitted through Coursera. Of the five sections that were submitted in this document, I wrote four of them. These sections include the answers to questions numbered 1, 2, 4, and 5.

**Team Project Status Update:** This document ended up being shorter in length than the "Team Project Proposal", but my contributions were still made. I wrote the paragraphs that we submitted for the questions numbered 1 and 2.

**Final Team Report:** Much to the same effect as the previous two documents, especially the "Team Project Proposal", I wrote the majority of the answers to the questions given in this document. Questions numbered 1, 2, 3, and 4 were answered by myself and then further refined by my team members.

**Project CTF:** The PCTF portion of the project is by far the largest section to cover as far as my contributions are concerned. To start, I created, maintained, and shared the initial code repository with each of the team members as soon as my team (Team 31) was formed. I enabled full access to the GitHub[1] repository for all team members so that there would be no barrier to entry. The initial commits that I made on the repository formed the core of the current script runner. I added an initial “example” script/module that was easily copy-and-pasted as needed. I wrote high-level instructions in the README.md file which explain how the program works and how it can be added to in the future. Once the design of our scripts was completed, I created the auto-runner, flag-submitter, target-info, vm-info, service-backup, and template-exploit modules. During the CTF event, I worked with multiple team members to create the backup-exploit, sampleak-exploit, and flaskids-exploit modules. These three modules were based on the foundation of the template-exploit module. Once the CTF event was over I committed those three exploit modules as well as the script.sh and script2.sh executable files that can be found in the root directory of the repository.

### 5. Lessons Learned

The work that I contributed to this project helped me to better understand some core fundamentals of networking and how networks can be penetrated without software systems noticing. My career is currently in the web development field of software engineering and I plan to, eventually, transition into the web security field. I can now say that I have a much deeper understanding of web-related hacking such as XSS, SQL injection, and broken authentication mechanisms. Before taking this course and participating in this project I thought that I understood web technologies enough to prevent these known classes of vulnerabilities, but now I know for certain that I know to safely implement protections against these types of vulnerabilities as well as others. As I have previously said, this project has taught me that I am capable of succeeding in the cybersecurity field and it showed me how fun and interesting cybersecurity is. I learned that my security interest is not waning and that I can directly use the newfound skills from this course in my day-to-day life as a software engineer. I am passionate about cybersecurity and because of this project, I am now signing up for and interested in completing more hacking challenges and competitions.

**Team #31:**
Captain: Chris Bilger
Hao Deng
Arkodeb Maity
Ganesh Natarajan
Louis Neira
James Russell

## References

[1] “Where the World Builds Software.” GitHub, github.com/.

[2] ChristopherBilg. “ChristopherBilg/Cse-545-PCTF-Team-31.” GitHub, 11 Feb. 2021, github.com/ChristopherBilg/cse-545-PCTF-team-31/blob/master/runner.py.

[3] ChristopherBilg. “ChristopherBilg/Cse-545-PCTF-Team-31.” GitHub, github.com/ChristopherBilg/cse-545-PCTF-team-31/blob/master/modules/net-analyzer.py.

[4] Shellphish. “Shellphish/Ictf-Framework.” GitHub, 31 Mar. 2020, github.com/shellphish/ictf-framework/blob/master/teaminterface_client/swpag_client/client.py.