Purple text on a black background

Description automatically generated

**Automation of Penetration Testing Procedures**

**A Capstone Project Report of MSIT 5910**

*Submitted by:*

Michael Couchman (C10111672)

*For the partial fulfilment of the requirements for the degree of*

**Master of Science in Information Technology**

*Supervised by:*

Dr. Shuchi Dhir

**Department of CS & MSIT**

University of the People, Pasadena, CA 91101, United States

**Term 1, September-November 2024**

# **Acknowledgement**

I would like to thank my supervisor, Dr. Shuchi Dhir, for supervision of this project, as well as the support towards successful completion. I would also like to thank Dr. Sanjay Bhargava for his oversight and for his outreach on Oracle Academy and for his work in the department. I would also like to thank my family and friends for their encouragement and support that helped me towards this goal.

# **Table of Contents**

Contents

[**Acknowledgement** 2](#_Toc181090248)

[**Table of Contents** 3](#_Toc181090249)

[**Abstract** 4](#_Toc181090250)

[**1.** **Introduction** 5](#_Toc181090251)

[**1.1 Problem Statement** 5](#_Toc181090252)

[**1.2 Overview of Solution** 6](#_Toc181090253)

[**2.** **Literature Review** 8](#_Toc181090254)

[**3.** **Methodology** 10](#_Toc181090255)

[**3.1 Requirement Analysis** 10](#_Toc181090256)

[**3.2 Software Development Lifecycle** 13](#_Toc181090257)

[**3.3 Project Life Cycle** 14](#_Toc181090258)

[**3.4 Programming Languages** 15](#_Toc181090259)

[**3.5 Software Testing** 15](#_Toc181090260)

[**4. Results** 19](#_Toc181090261)

[**4.1. Command Line View** 19](#_Toc181090262)

[**4.2. File Content View** 20](#_Toc181090263)

[**4.3. Database View** 21](#_Toc181090264)

[**4.4.Webpage View** 22](#_Toc181090265)

[**5. Discussion** 23](#_Toc181090266)

[**6. Conclusion** 24](#_Toc181090267)

[**References** 25](#_Toc181090268)

[**Appendices** 29](#_Toc181090269)

# **Abstract**

The purpose of this paper is to describe and report on the work done in MSIT5910 Capstone Project course, for partial fulfillment of the degree requirements for a Master of Science in Information Technology. The project was aimed at creating a solution for the substantial amount of time that is required in penetration testing procedures. The automation of different components through Bash scripting was utilized as a means of collecting data useful for penetration testing, appropriately formatting the data for both further use and permanent storage, storing the data in a database, and presenting the data in a reader-friendly format on a web page. All of this was coordinated and run consecutively to ease the amount of work required on the command line. A great deal of room for further development in future direction was discovered throughout the development, and this could also be applied towards different purposes. Overall, the work and findings demonstrated the power of automation towards cybersecurity applications and reinforce the importance of conceptualizing both offensive and defensive strategies to meet increasingly advanced threats.

1. **Introduction**

One of the modern shifts in cybersecurity relates to the potential for more advanced forms of cyber attacks to be used to achieve malicious aims. As time passes, and with the decreased complexity in implementing AI-based technology solutions, the potential for AI to be used as a cyberweapon is an increasing issue of concern in the world (Yamin et al., 2021). Guembe et al. (2022) note the sophistication of cyberattacks is increasing in recent years and posit that the adoption of AI-techniques in cyberattacks to evade cyberspace and cause greater damage without discovery is an inevitability on the part of cybercriminals. In addition to more advanced AI-threats, automation of penetration testing is another example of how newer methodologies are augmenting traditional practices. For example, Enoch et al. (2020) have proposed a novel automation framework for the purposes of penetration testing called HARMer.

These advances in emerging threats are what cued me develop my problem statement. The complex cyberthreat landscape, in addition to the challenges posed by automation, AI, or machine learning (ML), is also complicated by the expanding reach and attack surface that is available to cybercriminals. Cloud network attacks and the proliferation of IoT devices both also add to an increasingly complex environment from which to defend against attacks. In addressing the complexities, Mohammad and Surya (2018) highlight the strengths of using automation as a means of saving time and money, and it was essentially the extension of this idea which led me to my problem statement and proposed solutions for my capstone project.

## **1.1 Problem Statement**

At present, *much of the work of penetration testing procedures exists in a very linear and progressive way, and it can be a time-consuming process***.** It starts with research and passive reconnaissance and increasingly becomes more active and engaged throughout the attack. In the MITRE framework for example, this would proceed as follows: Reconnaissance, Resource Development, Initial Access, Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral, Movement, Collection (MITRE | ATT&CK, nd.). In a model such as the Cyber Kill Chain from Lockheed Martin, the steps progress as follows: reconnaissance, weaponization, delivery, exploitation, installation, command and control (C2), actions on objectives. The challenge in either model is that the process can be quite time intensive, with the attempt to gain access often taking very prolonged periods of time. One of the caveats of this is though that state-sponsored hacks, cyber warfare, and highly skilled cybercriminal actors, may often choose to adopt a “blanket” approach that seeks to exploit vulnerabilities in millions of systems to maximize the odds of success. As is common with an advance persistent threat (APT), such as those launched by state-sponsored actors, phishing or spearfishing campaigns, for example, will be launched broadly in attempt to find an entry point to a company’s network. This is to gain permanent backdoor access from which to operate (Airbus, n.d.).

Overall, the attempt to proceed through the components of a cyberattack, regardless of whether by ethical or unethical means, is something that can take a great deal of time if done manually. With respect to financing defensive measures, the cost can end up being significant. According to ZipRecruiter, average salaries for an ethical hacker differ by state in the US, ranging from ~$150K/year in New York to ~95K/year in North Carolina (ZipRecruiter, n.d). Regardless of which salary is being paid, the ability to effectively make use of automation to reduce the time required for aspects of penetration testing would be beneficial due to the associated costs.

## **1.2 Overview of Solution**

I am going to leverage my previous experiences with Bash scripting or Python to try to find ways to automate multiple aspects of the red teaming process so that they may allow penetration testing to be performed in a manner relevant to conditions that are of a larger scale than a single system, and therefore better emulate the conditions of real life. In addition to shortening aspects of the red teaming process through automation demonstrates how the process of red teaming may be augmented and improved with better technologies. This could include aspects of machine learning or AI as well. However, for the purposes and scope of this assignment, I suspect it would remain at simple automation. In addition to demonstrating how automation can improve penetration tests, it is possible to also elucidate better blue team strategies to defend against automated attacks and more advanced forms of cyberattack. Of what research was discovered on the topic of automation and cybersecurity, three relevant points are highlighted by Varga et al. (2022):

* Complicated cyber operations, both defensive and offensive in nature, are easy to execute regardless of resource level
* Those who possess the most advanced automated cybersecurity solutions have an advantage in the cyber domain.
* A few human work roles in cybersecurity are obsolete, deprecated, or simplified in complexity.

Overall, these points yield to a very relevant and impactful solution to my problem statement above. The end goal of the project is for the code to generate reports on reconnaissance for targets, as well as automatically perform subsequent steps in the attack process. In addition to the practical skills in cybersecurity, other aspects such as project management would also be relevant. Frameworks such as Scrum or Agile may allow for the practice of project management techniques throughout the project by dividing work into deliverables in a way that is consistent with weekly course content. In sum, the project would reinforce numerous aspects learned and allow for them to be reinforced in my final time in the program.

1. **Literature Review**

In recent years, the potential for autonomous, AI-driven cyberweapons has garnered significant public attention, while technological developments in the fields of artificial intelligence (AI) and cyberspace drive both inquiry and concern. As more nation-states come to possess these emerging technologies, the discourse around them intensifies. With respect state security, cyberattacks are now ranked among the most recognized defense risks. As such, NATO formally acknowledged cyberspace as an operational domain in 2016 (Shaoib, 2016). With the reliance upon digital technology and number of cybers attacks both steadily increasing over time (St. John, 2024), the importance of cybersecurity efforts shows no signs of slowing down. Such an importance is exemplified by the announcement of an additional $27.5B by the United States Government this year (Homeland Security Today, 2024). If we view things from business impact approach, the impact of cybercrime worldwide is estimated to be at $10.5 Trillion annually by 2025. Overall, the present state of the cybersecurity threat landscape is one that reminds of the both the impact of cybersecurity in general, as well as the need to keep pace with the potential for new types of threats such as those highlighted above.

One of the challenges in addressing this topic is the fact that much of the subject material is classified. However, it is proposed that the first notable offensive and autonomous weapons system to be AI-enabled would be deployed not on a physical battleground, but in cyberspace. It is also suggested that the United States would be most likely to first make use of the technology, with others like China thought to follow thereafter (Artificial intelligence and offensive cyber weapons, 2019). Part of these propositions and ideas can naturally cause prudent cybersecurity professionals to start thinking about ways to defend against such attacks. Part of this may also have to do with the seemingly unlimited number of ways that such attacks may be abused because of the complexity. For example, Bruno (2023) highlights concern over the use of AI-enabled cyberweapons to be able to target specifically groups disproportionately. Conflicts of national sovereignty, human rights, international law compliance, and the effect on civilians are several other concerns that were raised.

Overall, there is a vast potential for abuse of increasingly complex technologies, which has placed a considerable gap in defenses, as well as a gap in the literature on the topic. Unfortunately, with the array of legitimate applications that comes with technological advancements, there also concurrently exists opportunities for nation-state actors and cybercriminals to exploit. The gap in the literature is exemplified by a complete lack of statistics, for example, on the frequency of which attacks are occurring specifically by AI or automation. I could find no specific reporting or statistics that isolate AI-enabled attacks or attacks that have occurred using automation. In my opinion, three main factors infer a need to increase understanding in this area, and to place greater emphasis on defenses in this area. Firstly, AI, ML, and automation, have become more broadly available and that increased availability includes those who use the technology for malicious purposes. In other words, there is greater means of attack. Secondly, Internet of Things (IoT) devices increase the reach of technology in peoples’ daily lives, presenting a potentially greater attack surface and more opportunity. Third, global conflicts are at an all time high since World War II, providing an increased risk of state-sponsored cyberattacks. Together, these factors combine to underscore the importance of coming up with ways to conceptualize defenses against more advanced threats, such as those posed by automation or AI. This gap is what inspired the creation of this capstone project.

This capstone project aims to explore how automation can enhance cyberattacks by demonstrating the use of basic scripting in red teaming operations. It integrates multiple tools for open-source intelligence (OSINT) to automate reconnaissance tasks, store the collected data in a database, and present it in a user-friendly webpage format. This report format improves usability in red teaming by organizing information clearly. The project shows potential for further expansion, such as automating Metasploit attacks using gathered data, illustrating the broad possibilities of combining Bash scripting and Metasploit (TechXploits, 2021).

1. **Methodology**

This project sought to demonstrate the usefulness of automation in red teaming procedures towards reducing the amount of manual time required for testing. In this section, I will cover relevant aspects of the methodologies used towards project completion, as well as a requirement analysis. I will discuss the software development lifecycle (SDLC) approach that used, the project management applications relating to project, as well as the specific programming languages and software requirements that were utilized. I will also discuss the testing strategies used, validation and verification methods used in software testing.

## **3.1 Requirement Analysis**

The requirement analysis was a process by which the project manager decided upon the expectations and needs of a new product. The process involved repeated communication with project stakeholders as well as end-users to best define expectations, resolve any conflicts, and document all of the key findings (Simplilearn, 2024). Requirement analysis could be augmented by another type of analysis known as gap analysis. A gap represented a difference between an unmet need and a requirement. These could come in different forms, such as legal, performance, compliance, communication, and others (Fraculj, 2018). For example, if a legal standard or government regulation specified that the project needed to have a certain number of individuals with a certain type of certification, this would represent legal or compliance gaps. The gap analysis, therefore, had a reciprocal relationship with requirement analysis. It was important to note that requirement analysis was useful for constructing other components such as the project scope because the scope was the end goal and represented the sum of the requirements going into the final product (Project Skills Mentor, n.d.). Both elements took place in the planning phase of the project (Schindler, 2019).

As per Christian (2024), I went through the 4 steps mentioned in the requirement analysis process, detailed in the following sections.

Step 1: Identified key stakeholders and end users

For the project, the evident stakeholders that I identified were as follows:

* Myself, as the student acting in different roles, including aspects of project management, software development, research, and subject matter expertise in cybersecurity.
* University of the People, as the organization granting degree credentials.
* My supervisor, Dr. Shuchi Dhir, who had oversight of the MSIT5910 course.
* The end user could also be considered a stakeholder if the software was used.

The end user for whom this product/project was aimed was a cybersecurity practitioner with some basic knowledge of Linux operating systems and the command line interface (CLI).

Step 2: Defined the project goals

The goal of the project was to take the process of offensive penetration testing and reduce the amount of time required in the early phases of the process, such as reconnaissance and initial access (Dungarwal, 2022). The concurrent goals of the project were for me, as a student, to brainstorm, practice, and apply skills from courses throughout my degree program, and to demonstrate these toward the completion of the Master of Science in Information Technology degree. I also hoped to apply concepts learned from my previous certifications, including CompTIA Security+, ISC2 Certified in Cybersecurity, and Certified Scrum Master (CSM).

Step 3: Captured requirements from stakeholders

Many of the requirements for the project were satisfied by me, as the student operating under a variety of roles. Since the aim of the capstone project was to consolidate skills learned in previous courses I had taken at UoPeople, potentially relevant courses included databases, programming languages, managing projects and programs, and foundations in cybersecurity. From the other stakeholders, which included my supervisor and the University of the People as an institution, the requirements laid out included the structure provided by the MSIT5910 course. University of the People facilitated the Moodle platform and the organization of the course, which provided a means of communication and a common understanding that outlined the relationship between stakeholders.

Part of the requirements gathering technically began prior to the initiation of the MSIT5910 course, as my ideas for a problem and solution were approved in a previous course.

While the fundamental skills for the project were already in place due to previous work in MSIT courses, I still needed to specifically apply those skills to the project. To that end, I began researching some practical means of solving the problem. Preliminary research into the functional requirements suggested that I needed to review and apply the following skills:

* Bash programming, as a means of providing a CLI and integrating software solutions and tools. It would also be a means of generating any documentation, if applicable. Python knowledge, as well as a thorough knowledge of the cybersecurity process, was also a prerequisite for successful project completion.
* Web scraping knowledge, as a means of information gathering. This could come in the form of Linux CLI or languages such as Python, both of which were common for OSINT.

The program was designed to fulfill the following functions:

1. It must be determined if prerequisite packages or programs to be used throughout the project were installed.
2. It must utilize an array of OSINT tools to extract information from a user-defined target list to create customized reports on targets for surveillance. Web scraping could be used to extract key vulnerability information about targets and report that while omitting other information.
3. Relevant information on targets needed to be separated into temporary files or variables for subsequent reconnaissance work.
4. The information from step 3 needed to be piped into a vulnerability scanner, and the subsequent results needed to be appended to reports.

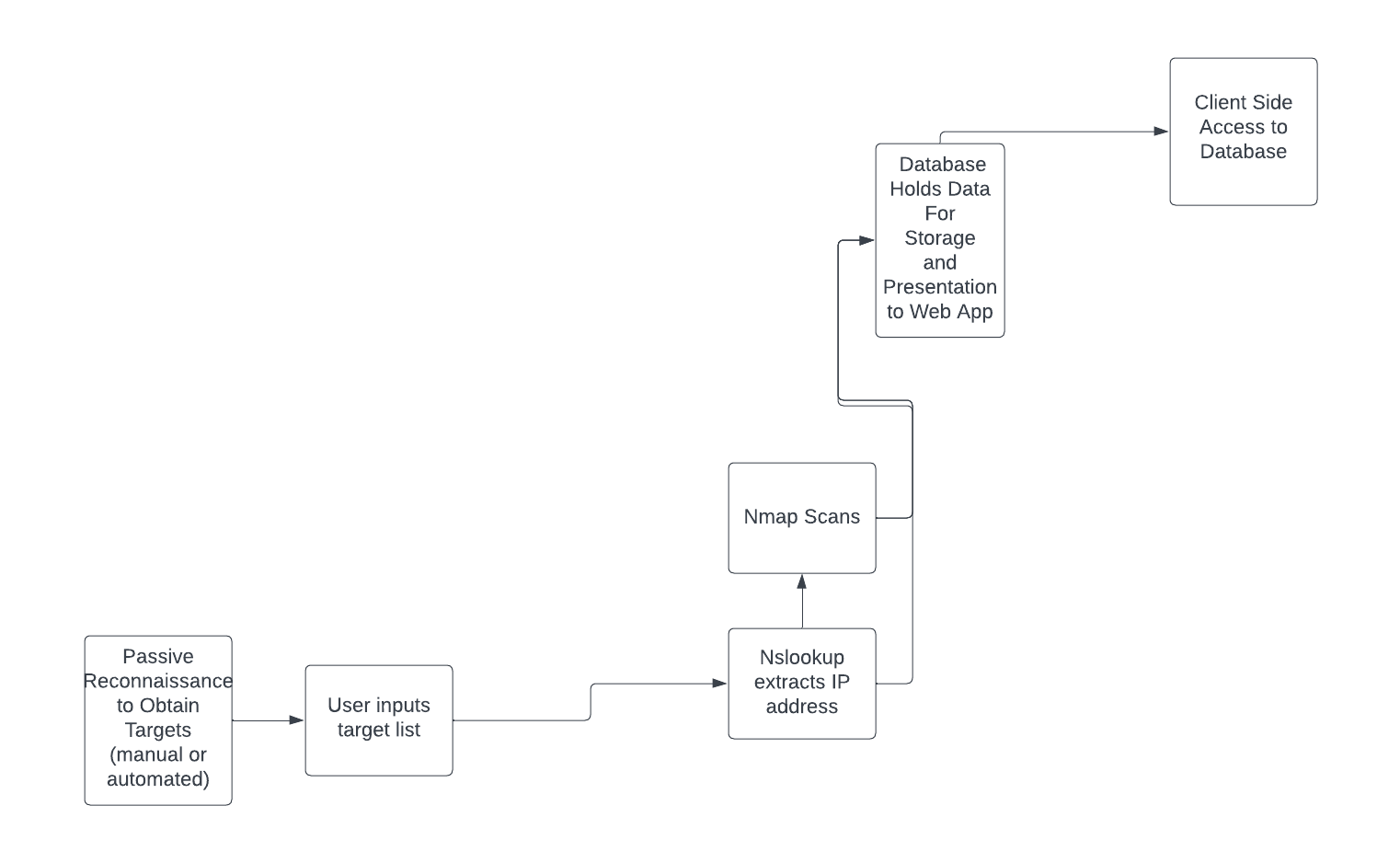
Step 4: Categorized requirements

As this section related to prioritization, feasibility analysis, impact analysis, and conflict resolution (Christian, 2024), I considered and discussed these topics. In terms of prioritization, the timeline laid out with the course curriculum ensured that each deliverable consecutively led to the next. For example, the problem statement and proposed solution were prerequisites to the requirement analysis. The timeline was believed to be feasible because it had been completed successfully by many students who had passed through the program.

In terms of impact analysis, it was important to maintain open communication and practice regular engagement in course forums to adapt to any unforeseen disruptions. If a disruption was believed to affect the proposed timeline, I would immediately contact my supervisor. Given the individual nature of the project proposal, it was not expected that conflict resolution would be a major concern. However, the university outlined conduct for students and faculty, as well as the processes related to conflicts should they arise.

## **3.2 Software Development Lifecycle**

The most applicable SDLC for this project would be the waterfall methodology. Whereas Agile methodologies emphasize flexibility and high levels of stakeholder engagement, the traditional Waterfall approach is more rigid in timeline and is more hands-off when it comes to stakeholder engagement (Hoory, 2024). The Waterfall approach is a better description for this project because the work goal was well planned out and steadily worked toward, instead of being broken down into iterative deliverables. Additionally, the fact that the timeline is inflexible on account on the academic calendar makes the approach more reminiscent of Waterfall. It is arguable that there are some of the projects that are relevant to Agile methodology as well, as it would depend greatly on the actions of the student. The end goal of the project was to achieve the software functionality that is demonstrated in the following diagram:



The process of the program begins with the provision of a list of target domains. This can be done either by some form of web scaping or manually input. For web scraping options, there are a wide variety of options. For example, curl is a command line tools that allow the extraction of web data on the command line. In either case, a list of targets is processed iteratively by Nslookup, which converts domain names to IP addresses. These are in turn iteratively processed by Nmap to perform two types of scans: a basic scan that displays information such as open ports, and a deeper vulnerability scan that reports on cybersecurity vulnerabilities. The information is then uploaded to an Apache Web Server database as part of the functionality of [Xampp](https://www.apachefriends.org/). This is achieved using SQL from the command-line interface. An application-program interface (API) is created utilizing PHP in joining together components of Bash scripts on the client machine, a database, and an accessible web page. An updated repository of code for this project is available [here](https://github.com/Mcouchman2024/MSIT5910_Capstone.git) at GitHub.

## **3.3 Project Life Cycle**

The project management lifecycle consists of five phases: initiation, planning, execution, monitoring, and closure (Kissflow, 2024). Alternatively, some models outline four phases: planning, build-up, implementation, and closeout (HBR Editors, 2016).

The initiation began in the previous semester with brainstorming ideas for a project evaluated in the MSIT5270 Foundations of Cybersecurity course. In the MSIT5910 Capstone Project course, the first two weeks focused on planning, which involved identifying the problem, proposing a solution, and conducting a requirements analysis, including creating a project timeline, Gantt chart, and stakeholder identification.

The execution phase commenced after the second week, utilizing ClickUp for project management. This phase included budget planning, human resources management, software specifications, security considerations, and progressive coding. Throughout, I applied newly learned course material and tracked progress, adjusting as necessary. The execution and monitoring phases occurred concurrently from week 3 onward, with a focus on validation and verification, and code uploaded to GitHub for version control.

The closure phase, taking place in the final weeks post-submission, includes feedback, retrospective analysis, and reflection.

## **3.4 Programming Languages**

The software development for this project was done using a combination of PHP, Bash, SQL, HTML, and CSS. PHP programming was an open-source server-side language typically used for web development (Chris, 2021), and it was considered a good choice for forming an API in this project. Since CSS and HTML formed a great deal of World Wide Web (WWW) technology, they were both considered useful and appropriate for styling the outputted data on the end user client side. SQL was chosen because it was the standard language for interacting with relational databases (Datacamp, 2018). The primary language used for this program was Bash due to its powerful and broad applications. In this project, Bash scripting was used to run commands, format input/output data, and pipe or redirect data through various desired pathways.

## **3.5 Software Testing**

There are three types of testing that exist based on the level of knowledge available to the tester about the internal components of the software or its interactions with the system. In black box testing, testers know nothing about the internal application, design, and internal configuration. White box testing is the opposite, where testers can directly observe the code and internal structures. Grey box testing is a compromise between the two, providing partial details to the tester (Savchuk, 2023). For this project, the type of testing I am performing is white box because I am also acting as the developer. For that reason, I have full access to the internal workings and code.

In addition to the types of testing mentioned above, there are additional ways of describing software testing (Pittet, n.d.). For example, testing can be done through automation or manually. It can also be described by methodology, such as:

1. Unit testing – Testing individual methods.
2. Integration testing – Testing how modules work together.
3. Functional testing – Focusing on the business requirements of the software.
4. End-to-end testing – Examining the complete application environment.
5. Acceptance testing – Verifying that a system meets business requirements while the whole program is running.
6. Performance testing – Testing the software under various workloads.
7. Smoke testing – Basic and quick tests to ensure major components are working as expected.

The types of tests that have been conducted so far would likely also match the descriptions of smoke testing because they are occurring “on the fly” to ensure that the main components of the programs are structured and functioning as intended.

The unit testing observes the functionality of components. As such, my testing focused on each component in a standalone state, observing it for proper functionality first. Then, after ensuring each component performs its function(s) as intended, I am reasonably assured that the successive interaction between modules becomes feasible. Following unit testing, I performed several integration tests to ensure that the entire program ran as intended with a variety of test cases. The following test cases exemplify the process:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Preconditions** | **Expected** | **Expected Result** | **Actual Result** | **Status** |
| A1 | Storage of data from reconnaissance steps | Progress through nslookup to nmap and other commands | Data must be able to be output | Data should appear in nsl\_results.txt, nm\_bas.txt, nmres.txt | Data is properly formatted and stored in files for subsequent retrieval | Pass |
| A2 | Extract IP address from domain name provided to Nslookup, processed and subsequently passed to Namp | Domain name must be present in target list file.  Output IP must be provided to Nmap | Domain name should run through nslookup and relevant IP address should be extracted. Then, this information should be passed nmap | An IP address should be present in nsl\_results.txt and usable for nmap | Domain name failed, likely due to firewall, block, or network failure. No addition is added to nsl\_results.txt. Nmap is unable to run. | Fail |
| A3 | Extract IP address from domain name provided to Nslookup | Domain name must be present in target list file | Domain name should run through nslookup and relevant IP address should be extracted | An IP address should be present in nsl\_results.txt after running | Nslookup fails and outputs error message that is unsuitable for subsequent program steps. The program is unable to proceed properly. | Fail |

With respect to software testing cases above, additional code was entered to be handle exceptions to proper function. For example, if the Nslookup is unable to resolve an IP address, the program will still be able to proceed onward. The subsequent revisions that were made following software testing were to remove two components that were not felt to be necessary for the completion of the project. The first aspect that was removed was a WhoisDNS lookup component that sought to process and output DNS registry information. This component was scrapped because it was felt to be somewhat superfluous for the main goal of the project, which was focused on automating components of an attack. While registration information could foreseeably be used in domain hijacking (Tunggal, 2024), it is less relevant for this type of demonstration. The other aspect that was abandoned was the idea of a more detailed view for the website. This was because the condensed information in the table provided were felt to be sufficient to meet the concise goals of the project.

**4. Results**

The results will be demonstrated through screenshots and commentary to demonstrate all components of program. These are as follows:

1. Command line view
2. File content
3. Database view
4. Webpage view

## **4.1. Command Line View**

A screenshot of a computer

Description automatically generated

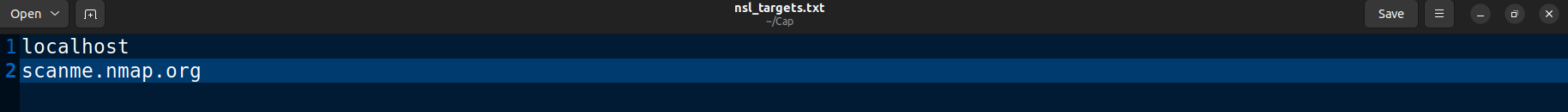
As can above, the program proceeds successfully through a software check, nslookup, and ping, before proceeding to nmap scans. These appear below.

A screenshot of a computer

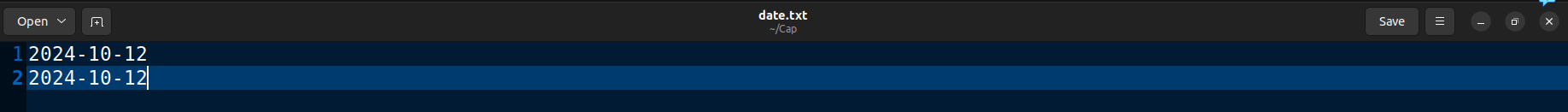
Description automatically generated

Nmap scans complete and the subsequent data is combined and formatted appropriately for database upload. In this case, several vulnerabilities have been detected. The database upload then occurs

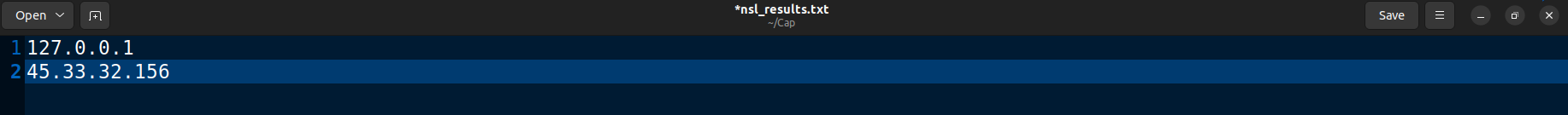
## **4.2. File Content View**



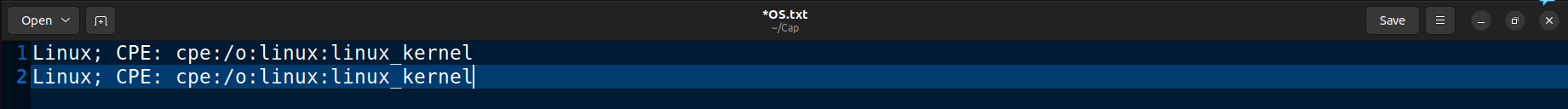
As can be seen, two targets are specified in the targets list. Note that these two targets also correspond to output data on the webpage displayed for the end user.



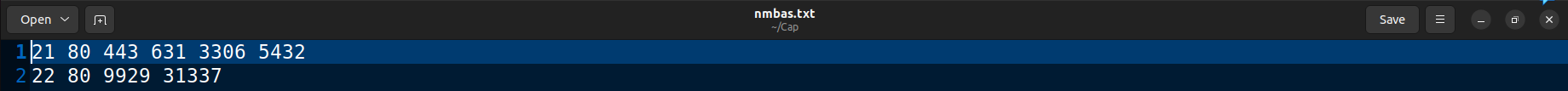
The data of the test is stored in “date.txt”



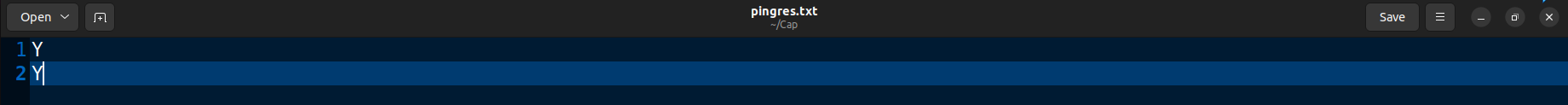
Subsequent output of the nslookup can be viewed in “nsl\_results.txt”, which is used Nmap scans.



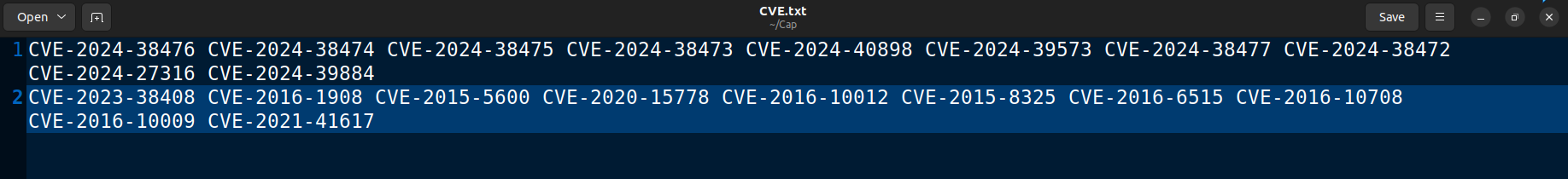
Operating system information discovered by Nmap scans are stored in “OS.txt”



Information on open ports discovered by Nmap scans are stored in “nmbas.txt”



Information of whether the target responds to ICMP Ping is stored in “pingres.txt”



Information on vulnerabilities discovered is processed and stored in “CVE.txt”

## **4.3. Database View**

**A screenshot of a computer

Description automatically generated**

## **4.4.Webpage View**

A screenshot of a computer

Description automatically generated

As can be seen, the webpage view demonstrates the final output of the program run on the command line. The values have been retrieved and neatly presented in a stylized page for the user.

Recapping the results above, as can be seen throughout the screenshots, the program initiates on the command line and progresses through its different components, storing information along the way in way that is suitable for upload to the database. The program uploads the data into the database, which is then able to be displayed in a human-friendly format (web page).

**5. Discussion**

Overall, I felt that the project progressed satisfactorily. I was glad to see everything generally coming together with all the project components functioning as intended towards their initial goals. I believe that the removal of the elements mentioned above were best to prevent scope creep and ensure that a demonstration was provided towards a specific goal of automation of different components. I found this to be a bit difficult because I noticed a tremendous number of areas in which this project could be expanded or scaled up. For example, I had originally intended to incorporate a component that integrated this data to automatically perform Metasploit attacks. There are also ways in which web scraping could be automated and included into this, further expanding a workflow outward. The possibilities for future directions for this type of project seem quite broad, and they could also be applicable for multiple purposes as well. For example, it could take more of a direction towards defensive countermeasures, or it could be adapted more towards open-source intelligence gathering. It could also be adopted to format this data into a report for informing policies, updating security personnel, and much more. I resisted the urge to delve into this, despite the opportunity being there. This is because of the time and resource restrictions inherent to a single-person project run over a single academic term at the University of the People.

It is hoped that this basic demonstration provides a glimpse of how automation may drastically cut down time in offensive operations, but it should also be noted how equally this may apply to defensive technologies as well. As I take note of the plethora of ways in which this small project may be scaled up to achieve more powerful results and applied toward different applications, I suspect that this effect would only be amplified greater with driving power of big data, AI, or largescale automation.

**6. Conclusion**

The successful completion of this project demonstrates the capacity of automation to be able to reduce the amount of time required for penetration testing. While it is merely a proof of concept for educational purposes, and to demonstrate several skillsets learned throughout the course of the Master of Science in IT program, the potential for this type of application to be extended is feasible and could be done in a way that is suited towards a wide variety of applications. As cybersecurity professionals continue to face increasingly advanced threats, such as those posed by automation or AI, it is important to be able to investigate means by which both offensive and defensive cybersecurity measures may occur through the same technologies. While the application of this project was centered around being educational in nature, the speculation about AI-enabled cyberweapons released into cyberspace is likely going to be something to contend with in coming years, and this should prompt those in cybersecurity to start thinking about ways by which to protect against more complex forms of attack.

(4828 words)

**References**

Artificial intelligence and offensive cyber weapons. (2019). *Strategic Comments*, *25*(10), x–xii. https://doi.org/10.1080/13567888.2019.1708069

Adobe Communications Team. (2022, March 18). *Project Monitoring and Controlling Phase.* Adobe Experience Cloud Blog. https://business.adobe.com/blog/basics/monitoring-controlling#:~:text=Project%20management%20monitoring%20and%20controlling,Staying%20within%20budget

Bruno, Z. (2023). *AI-Enabled Cyber Warfare: The Future of Cyber Conflicts.* Academia.edu https://www.academia.edu/100903433/AI\_Enabled\_Cyber\_Warfare\_The\_Future\_of\_Cyber\_Conflicts

Chris, K. (2021, Augst 30). *What is PHP? The PHP Programming Language Meaning Explained.* freeCodeCamp. https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/#:~:text=PHP%20is%20mostly%20used%20for,show%20it%20in%20the%20terminal.

Christian, A. (2024, April 10). *Requirements Analysis Process and Techniques*. Requirement. https://www.requiment.com/requirements-analysis-process-and-techniques/

Clickup [Computer Software]. (2017). Retrieved from: https://app.clickup.com/

Datacamp. (2018, October 15). SQL *Server, PostgreSQL, MySQL… What’s the difference? Where do I start?* https://www.datacamp.com/blog/sql-server-postgresql-mysql-whats-the-difference-where-do-i-start

Dungasrwal, V. (2022, September 22). *The Five Stages of Red Team Methodology.* SecurityBoulevard. https://securityboulevard.com/2022/09/the-five-stages-of-the-red-team-methodology/

Enoch, S. Y., Huang, Z., Moon, C. Y., Lee, D., Ahn, M. K., & Kim, S. K. (2020). HARMer: Cyber-Attacks Automation and Evaluation. *IEEE Access.* https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9142179

Fraculj, M. J. (2018, July 31). *Conducting requirements and gap analysis in projects.* Sysco Middleware Blog. https://blog.sysco.no/business/analysis/Conducting-requirements-and-gap-analysis-in-projects/#:~:text=Generally%2C%20requirements%20can%20be%20business,and%20needs%20to%20be%20remediated.

Guembe, B., Azeta, A., & Misra, S. (2022). The Emerging Threat of Ai-driven Cyber Attacks: A Review. *Applied Artificial Intelligence*. 36(1). https://www.tandfonline.com/doi/full/10.1080/08839514.2022.2037254

Kissflow. (2024, August 13). *5 Phases of Project Management -A Complete Breakdown.* https://kissflow.com/project/five-phases-of-project-management/

HBV Editors. (2016, November 3). *The Four Phases of Project Management.* Harvard Business Review. https://hbr.org/2016/11/the-four-phases-of-project-management

Hoffman, B. G. (n.d.). *What is red teaming?* RedTeamThinking. https://www.redteamthinking.com/what-is-red-teaming

Hoory, L. (2024, May 31). *Agile Vs. Waterfall: Which Project Management Methodology Is Best For You?* Forbes. https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/

Homeland Security Today. (2024, March 13*). US Government Sets Aside $27.5B for Cyber Security.* https://www.hstoday.us/subject-matter-areas/cybersecurity/us-government-sets-aside-27-5bn-for-cyber-security-spending/

Pittet, S. (n.d.). *The different types of software testing*. Atlassian. https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing

Project Skills Mentor. (n.d.). *Project Requirements: Scope, and Deliverables: The Purpose of a Project*. https://projectskillsmentor.com/project-skills/project-requirements-scope-and-deliverables#:~:text=The%20Scope%20is%20the%20goal,than%20just%20the%20core%20product.

Savchuk, Y. (2023, January 24). *A Guide To: White Box, Black Box, and Gray Box Testing.* Careerist. https://www.careerist.com/insights/a-guide-to-white-box-black-box-and-gray-box-testing

Shoaib, M. (2016). AI-Enabled Cyberweapons and implications for cybersecurity. *Journal of Strategic Affairs.* 9-37. https://www.semanticscholar.org/paper/AI-Enabled-Cyber-Weapons-and-Implications-for-Shoaib/705c17ce88d1c4d88b14159abd617314afab461e

Sobers, R. (2024, September 13*). 157 Cybersecurity Statistics and Trends [updated 2024].* Varonis. https://www.varonis.com/blog/cybersecurity-statistics

St. John, M. (2024, August 28). *Cybersecurity Stats: Facts and Figures You Should Know.* Forbes. https://www.forbes.com/advisor/education/it-and-tech/cybersecurity-statistics/#:~:text=As%20the%20globe%20becomes%20more,%25%2C%20surpassing%20the%20previous%20record.

TechXploitz. (2021, June 8). *Automate Metasploit With Your Own Bash Script [Tutorial]* [Video]. YouTube. https://www.youtube.com/watch?v=-kc-9xxcurs

Tunggal, A. T. (2024, September 16). *What is Domain Hijacking?* UpGuard. https://www.upguard.com/blog/domain-hijacking

Varga, S., Sommestad, T., & Brynielsson, J. (2022, August 1). Automation of Cybersecurity Work. *Artificial Intelligence and Cybersecurity.* 67-101. https://link.springer.com/chapter/10.1007/978-3-031-15030-2\_4

Vision of Humanity. (2024, June 11). *Highest number of countries engaged in conflict since World War II*. https://visionofhumanity.org/highest-number-of-countries-engaged-in-conflict-since-world-war-ii/

Yamin, M. M., Ullah, M., Ullah, H., & Katt, B. (2021, March). Weaponized AI for cyber attacks. *Journal of Information Security and Applications*. 57. https://www.sciencedirect.com/science/article/pii/S2214212620308620?casa\_token=6arg-y6I9O4AAAAA:npUluJIdyXM9gMGUz-I\_GOyWDszXTGELoKBlSFgNWqmxyjhoOpHrrm32dXVG4UOj08oKSnvkNv50

ZipRecruiter. (n.d.). *What is the Average Ethical Hacker Salary by State.* https://www.ziprecruiter.com/Salaries/What-Is-the-Average-Ethical-Hacker-Salary-by-State

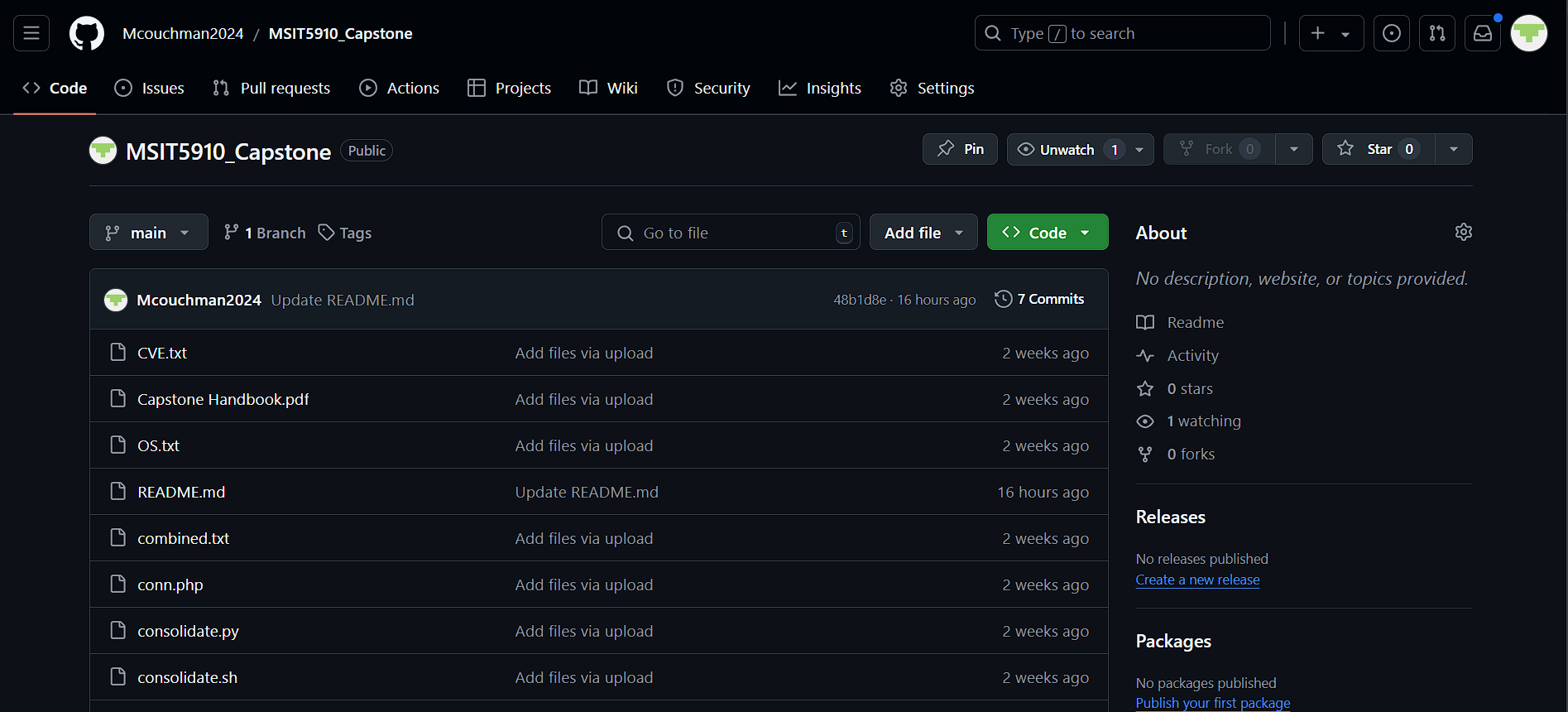
**Appendices**

**Appendix A.**

Software Repository

A software repository was created for this project can be found at the following link. The repository contains the script files, purpose of the project, and key documentation used to guide the development of the project.

<https://github.com/Mcouchman2024/MSIT5910_Capstone>



**Appendix B.**

Video Presentation Files

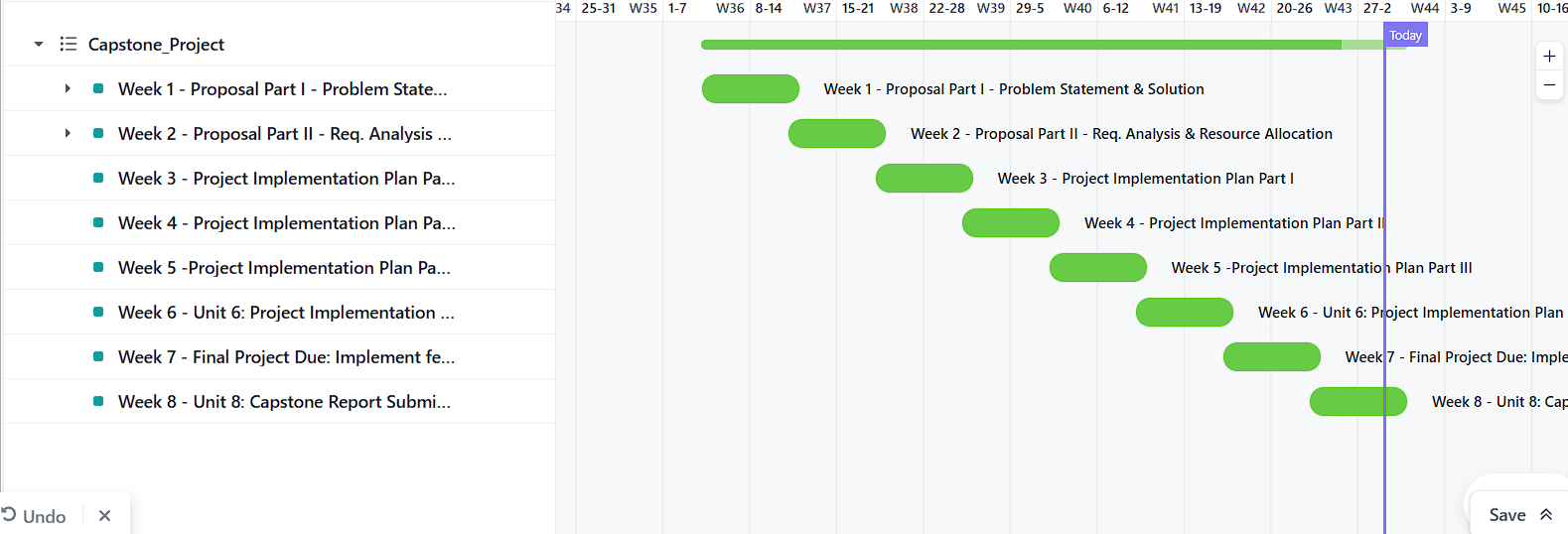
A video presentation file can be found at the link that follows. The presentation provides further information about the project, such as a demonstration of usage, purpose of the project, as well as other components already detailed in this document.

<https://go.screenpal.com/watch/cZ6UcJncqe4>

**Appendix C.**

ClickUp Gantt Chart

A Gantt chart for this project was created using Clickup. The following figure demonstrates the use of this feature for the purposes of project management throughout the course of the project up until the time up submission.

****

**PART II: Course Reflection**

Week 8 Capstone Course Reflection Assignment

Michael Couchman

University of the People

MSIT5910 – Capstone Project

Supervisor: Dr. Shuchi Dhir

Date: 2024-10-29

**1. Explain the rationale behind choosing this specific topic for your capstone project.**

The rationale behind choosing this topic was two factors. The first of this was some previous training and experience in cybersecurity and scripting. The second was because I foresee a gap in this area of technology. When I see the way in which automation, machine learning, and AI are rapidly driving technology, I foresee there will be opportunities in cybersecurity to consider the implications from both an offensive and defensive perspective.

**2. How does your capstone project align with the Course Learning Outcomes (CLOs)?**

- CLO 1: Apply the concepts of information technology, computer science, and allied disciplines to solve complex problems.

This CLO applied because the problem was complex, and the solution required the implementation of several different areas of technical proficiency. These included aspects of software development (full stack), cybersecurity, programming languages, and project management.

- CLO 2: Design a solution using emerging tools and technologies to solve a business problem.

The solution was designed to meet a very specific problem, and the tools used are relevant to modern business solutions in the field of IT. For example, both databases and cybersecurity measures to protect business assets are of vital importance in driving many billions of dollars of revenue. The topic of addressing automation and more advanced cybersecurity threats is also a contemporary challenge that is need of consistent forward-looking solutions.

-CLO 3: Evaluate the solution in context of the objectives defined for the project.

The solution implemented meets the goals set out because it effectively demonstrates the ability of automation to reduce time required for certain aspects of penetration testing. In doing so, the project also demonstrates means by which automation technology can potentially be used for offensive and aides in conceptualizing defensive strategies as well. As such, the main objectives in the proposed solution were met with this project.

-CLO 4: Analyze user needs for system design and development, based on a business problem

This objective was met because the product brought together the following components towards solving a business problem: database, API, web site, command-line interface. Each component was coded and tested, and then ensured to function together as intended towards solving a business problem. The final product functions as intended in saving time.

-CLO 5: Critically analyze the ethical considerations when managing IT systems.

This objective was considered and met by ensuring that the user is frequently warned about the purpose of the software for educational purposes, as well as for the sole responsibility upon the user for acceptable use. The final software product also did not incorporate

-CLO 6: Demonstrate the ability to write well-organized arguments supported by high-quality, credible, relevant sources.

The writing was keeping with the outlines specified and supported by numerous recent, high-quality references/sources.

**3. How do you believe your capstone project can make a meaningful impact in the field of IT?**

I believe that the project can make a meaningful impact by demonstrating the ways in which automation may can reduce redundancy in penetration testing processes, and by encouraging others to increasingly be aware of the potential for automation to be used for both defensive and offensive cybersecurity. As the number of cyber-attacks has substantially increased over recent years (Novikava, 2024), it is also hoped that the applications of the project and increased awareness it seeks to promote will both help to create a safer digital world.

**4. How can the organization you work for benefit from the implementation of your capstone project?**

An organization would potentially benefit from the implementation of this project in a few different ways. The first would be a realization of the power of automation in engaging in cyber attacks or penetration tests. From a perspective of defending one’s own assets, it may be useful to use a project like this to conceptualize ways of defence. For example, in this case, a single IP address is launching the attacks and ways to mitigate this attack could simply include blacklisting the IP address on a firewall. However, since the potential is there for these to sorts of scans to be scaled up over multiple devices, what would I do in such a circumstance? These sorts of questions are useful for driving a thinking towards defences against increasingly complex attacks. Another benefit that would directly be applicable would be a reduction in the amount of time required in penetration testing. Alternatively, software such as this could be easily repurposed to perform other scans or even perform Metasploit attacks.

**5. What challenges did you face during the development and implementation of your capstone project and how did you overcome them?**

One of the constraints I ran into in the development and implementation of this project was time. A project like this has enormous potential to be able to achieve more and more functionality and this challenged me to continually refocus on the project goals and prevent scope creep. I found this to be quite difficult because it encompasses aspects of time management, but it also required me to practice good project management skills dynamically throughout the project to cut potential development areas that would take too much time while still incorporating the aspects important to achieving the goals. I also found this to be challenging because a lot of ideas came to me on the fly about what really impressive options may be useful to be included, but further inquiry demonstrated that the time required would not make it feasible.

**6. What are the future directions or potential areas for further improvement related to your capstone project?**

Throughout the progression of my capstone course, I saw a lot of different ways in which the project could be expanded. While this project only covered a couple elements of offensive penetration testing procedures, it is possible for that to be extended through to an entire workflow that passes through all elements of the cyber kill chain: reconnaissance through to actions on objectives (Cyber kill chain. Lockheed Martin, n.d.). It is also possible for the same integration and automation to be applied to different aims. One example could include the automated collection, process, database upload, and display of various OSINT scans or cyber threat intelligence (CTI) sources. It is also possible for the output of this program to be directed into a framework like Metasploit to initiate the attacks and it is possible to monitor the results of such procedures through more advanced programming.

**References**

*Cyber kill chain®*. Lockheed Martin. (n.d.). https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Novikava, A. (2024, July 3). *The evolution of cyber threats: looking back over the past 10 years.* NordLayer. <https://nordlayer.com/blog/evolution-of-cyber-threats-over-10-years/>

**EasyBib Plagiarism Score:**

A close up of black text

Description automatically generated