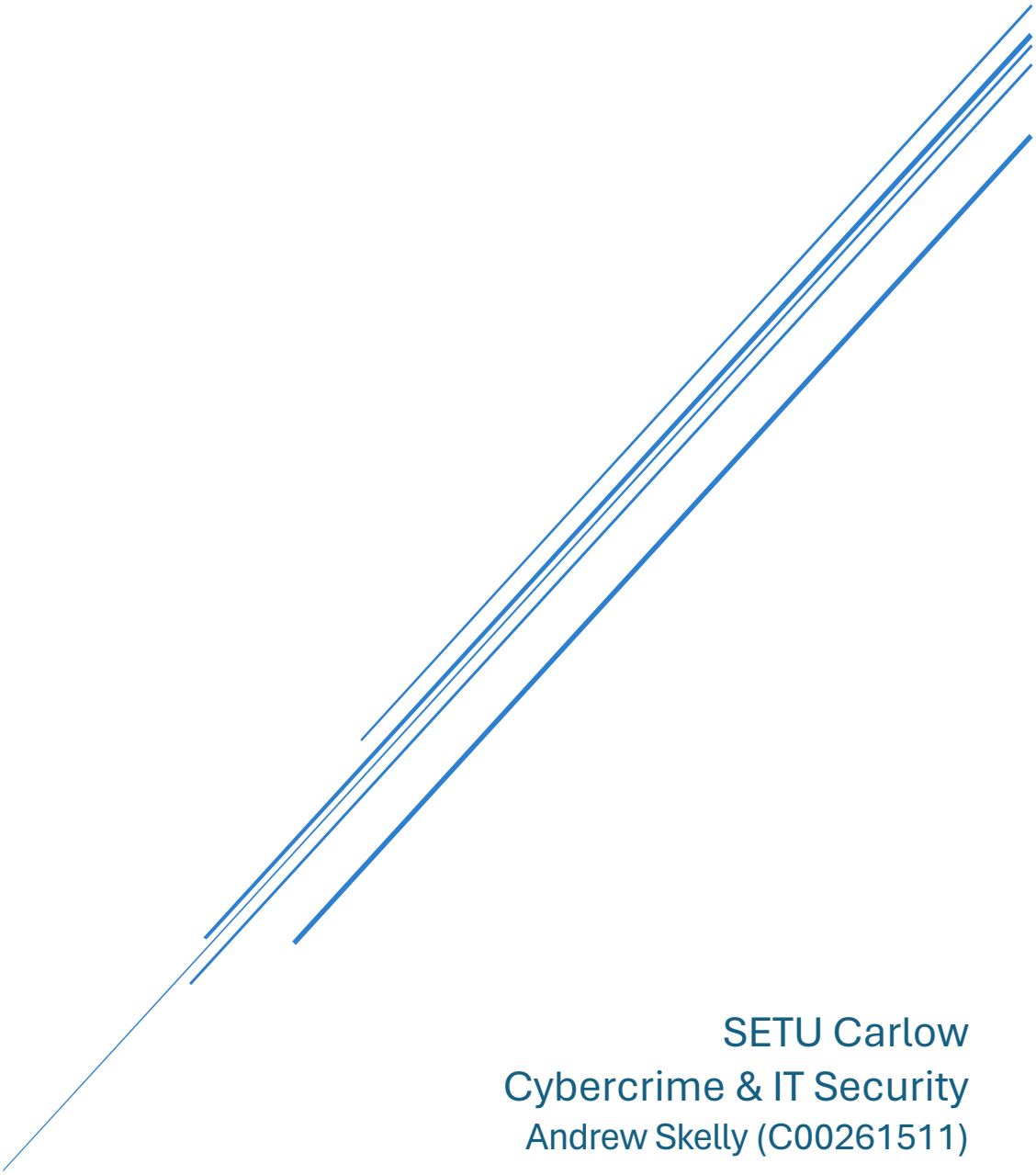
ASSET IDENTIFICATION AND RISK ASSESSMENT FOR SMES

Final Report - My Journey of Completing My FYP



SETU Carlow
Cybercrime & IT Security
Andrew Skelly (C00261511)

Contents

Abstract

This document aims to explain, in detail, the experiences that I gathered while completing my project, the things I learned and did well, and the things that I believe I could have done better.

In the current, modern-day world, cybersecurity is becoming more of a concern for Small to Medium sized Enterprises as they are being hit by an increasing number of cyber-attacks. This is due to a lack of reasonably priced and simple-to-use tools that allow these companies to easily identify their assets, the vulnerabilities they have, and how to remediate these threats.

Apollo Defender is a network vulnerability scanner that is intended to help these Small to Medium sized Enterprises combat the ever-increasing threat of cybercrime. It does this by alerting the users to any vulnerabilities that are present on their network and giving them the information they need to resolve the threat and keep their environment safe from compromise.

Acknowledgements

I would like to thank Christopher Staff, my project supervisor, for his excellent and consistent guidance and commitment to helping me create as best an application as possible.

The Aim of this Project

The original vision of this project was to create a tool that could be used by a Small to Medium Sized Enterprise (SME) that would help improve their network security greatly by automating the task of searching for vulnerabilities on hosts that are on the SME's network. It was planned to do this by using Nmap to scan the target network and gather information on the hosts. This information would then be passed to the National Vulnerability Database API which would search for Common Vulnerabilities and Exposures (CVE's) based on the information gathered by Nmap. Any vulnerabilities detected would then be recorded and displayed to the user.

The information gathered about hosts and their vulnerabilities would then be compiled into a document which could be scheduled on a weekly, monthly, or user defined basis. Any high-risk vulnerabilities would cause an alert to be sent to the email of the user if an email is provided.

The project was intended to be designed with a security by design mindset, where the application would be built from the ground up with a focus on security, due to the nature of the information being gathered by the tool. People may question the security of the storage of the file that contains the vulnerabilities, but the approach that has been taken is that this file will be stored on only one machine (the machine that the program has been run on) and unless an attacker has access to that machine, then the file and its contents will be safe.

Overall, the tool, which I have named Apollo Defender, would be a valuable tool for SMEs that do not have access to enterprise level software or a large security team, but still have a large enough network where one admin could not easily handle all possible threats.

What I Learned While Completing this Project

The main thing that I learned while completing this project was the programming language Python. While I had an interest in Python and had a small amount of knowledge about how it worked before starting this project, I did not have enough knowledge to write a program like I have created. Through the use of YouTube tutorials, guides, and tips combined with StackOverflow and tinkering around with the language myself, I gained a large amount of knowledge about the usefulness of Python and some of the things that it is really powerful at.

One of the most useful things that I feel like I have learned while learning Python is something called list comprehension. List comprehension is the practice of using different algorithms and sorting methods to loop through a list or dictionary of items and pull out needed data. This came in very useful when I was parsing through JSON files as a JSON file is essentially a large list filled with nested lists and dictionaries. The knowledge that I gained about list comprehension benefitted me greatly when figuring out how to effectively parse JSON files. *(Fig. 1)*

```
return {
    address['@addr']
    for host in data['nmaprun'].get('host', [])
    for address in host.get('address', [])
    if address['@addrtype'] == 'ipv4'
} or {
    self.get_host_addresses(hosts)
    for hosts in [data['nmaprun'].get('host', [])]
    if not isinstance(data['nmaprun'].get('host', []), list)
    if not self.ip_addresses
}
```

Fig. 1 shows list comprehension being used to find valid and active IPs.

Another thing that greatly benefitted me was learning the power of Python libraries. Python libraries are pieces of code that have been developed by other people for completing tasks that would normally be very time consuming to write code for. Libraries in Python are a great help for this as it means that I can just call functions from the libraries that I have imported, allowing me to complete complex tasks like converting XML files to JSON files with one command rather than having to write the code myself to do this.

A third thing that I have learned in Python while completing this project was the use of one of Python's inbuilt GUI frameworks called PyQt. PyQt is a simple yet powerful GUI framework that allows for many different elements such as radio buttons or file selection

buttons to be added to the window with little effort. Learning about this framework made it easy for me to develop a GUI that was intuitive to use and simple for a user with some experience and understanding of their network to pick up.

Moving away from the topic of Python, another thing that I learned while completing this project was the inherent risks involved in creating a program that is designed to scan a network for vulnerabilities. Of course, in safe and responsible hands, this program will be a great use to SMEs in helping them identify vulnerabilities in their network. However, in the hands of a threat actor, this program could be used to identify vulnerabilities in a network and exploit these vulnerabilities to gain a foothold. This is a consideration that needed to be accounted for when developing this application. I achieved this in one way by only allowing the user to use a predefined set of scan types. The program allows the user to select from a normal scan, and aggressive scan and a stealth scan, which have all been written into the code and cannot be changed. By not letting the user specify their own command fully, this reduces the risk posed should threat actors get a hold of this software.

One final thing that I feel like I learned a lot about during the development process was time management. I think that, although I did not necessarily manage my time perfectly, I made a good attempt to and learned a lot about time management of a project with a lot of individual parts like this one. There were many things that had to be balanced throughout the development of this project, such as the initial project specification, journals, the research report , and the presentation. All of these needed to be worked on at the same time as I was developing the code for the application or learning Python so I could code the application. Throughout the year, I worked on two or even three different documents for different project components at the same time in order to meet deadlines. In this regard, I learned a lot about the way that I work with the time that I am given. I have discovered that if I have several objectives to accomplish, each with a different but close deadline, I will usually prioritise the task with the closest deadline while still beginning or continuing work on the secondary or tertiary task, though to a lesser degree than the main assignment . I feel like this helped me keep myself motivated mentally as it felt like I was always making progress in at least one area, even if another area was causing me trouble. This all stemmed from the way that I managed my time during the development of this project.

The other side to this was that at times, I feel like I could have made better decisions when it came to time management. Sometimes, tasks like the presentation or one of the reports took up more time than they were supposed to take, compared to the time I should have spent developing the project's code.

Overall, when it comes to time management, I feel like there were some things that I did well and some things that caused me to fall behind in some areas. I definitely learned a lot about the way that I manage my time and I have realised and learned about ways that I can better manage my time on large projects like this in the future.

Decisions I Made That I Was Happy With

One of the biggest decisions that I was happy with was the choice I made to move from attempting to use an API based vulnerability detection mechanism to a local based vulnerability detection mechanism. From the inception of this project, I believed that using cloud-based systems would allow much easier accessibility to users, but as time went on into the development process, I come to see the flaws and possible issues that this approach could take. For instance, I recognised relatively early on, that choosing a cloud-based strategy would mean that, in the event that the cloud storage service was compromised, any files or tables containing sensitive information may potentially be leaked at any time. I believed that it would be better, given the security by design mindset that I wanted to take with this project, that I move to a more local approach. This means that any files that contained sensitive information would only be stored on one user account on one PC in the network. Consequently, only the person that has access to this user account would be able to view and share the files to people who they deem necessary.

Another reason that I am happy with this decision is because I have never used or communicated with any sort of API before, even though I had a relatively good understating of how an APIs worked. The challenge of creating a whole GUI application, figuring out how to correctly parse all the JSON information, send this information to an API and finally interpret this API response and display this information to the user would have been too much to try to develop in the small amount of time that I had. Given the implications of trying to develop an effective way to communicate with this API, I felt that it would be best to move to a more local approach. I believe that doing this saved me a great deal of time overall in terms of development time, so this is a decision I'm happy I made at a relatively early stage.

Another decision that I am extremely happy that I made was the decision to use Python as the language that this project would be developed in. The main reason that I am happy about this decision is because Python is a language that I had not learned at that point in any modules in college, despite it being one of the languages that I was most excited to learn. This is due to the simplicity of its syntax as well as the power possessed by its many different libraries alongside its generally powerful features such as list comprehension. This is because of its generally powerful features, including list comprehension, and the power of its different libraries, in addition to its simple syntax. I feel like Python will be vital to me having a successful career later on in the future, as it is a language that it very widely used in the cybersecurity world for automating tasks and running scripts.

Python was also very useful for this project as it made looping through all the lists and dictionaries that were contained in the JSON objects much easier than it would have been in other languages. With a fraction of the code needed in other languages, I was able to perform tasks like opening and writing to files. All of this allowed me to make the code of this project relatively compact given the complexity of the tasks that were being completed. Since Python was developed to be simple to read, almost like reading a book, this also lends itself to the ease of reading and understanding the code, should you be a person reading the code for the first time.

What I Achieved

During my time implementing this project, I achieved many things. One of the main things that I have achieved was creating a functional and simple UI that made it simple for the user to start a scan. I created a UI using PyQt, one of the built-in GUI frameworks in Python, so that anyone with some networking knowledge could open the application and do a scan quickly and easily without needed to manually verify databases (*Fig. 2*).

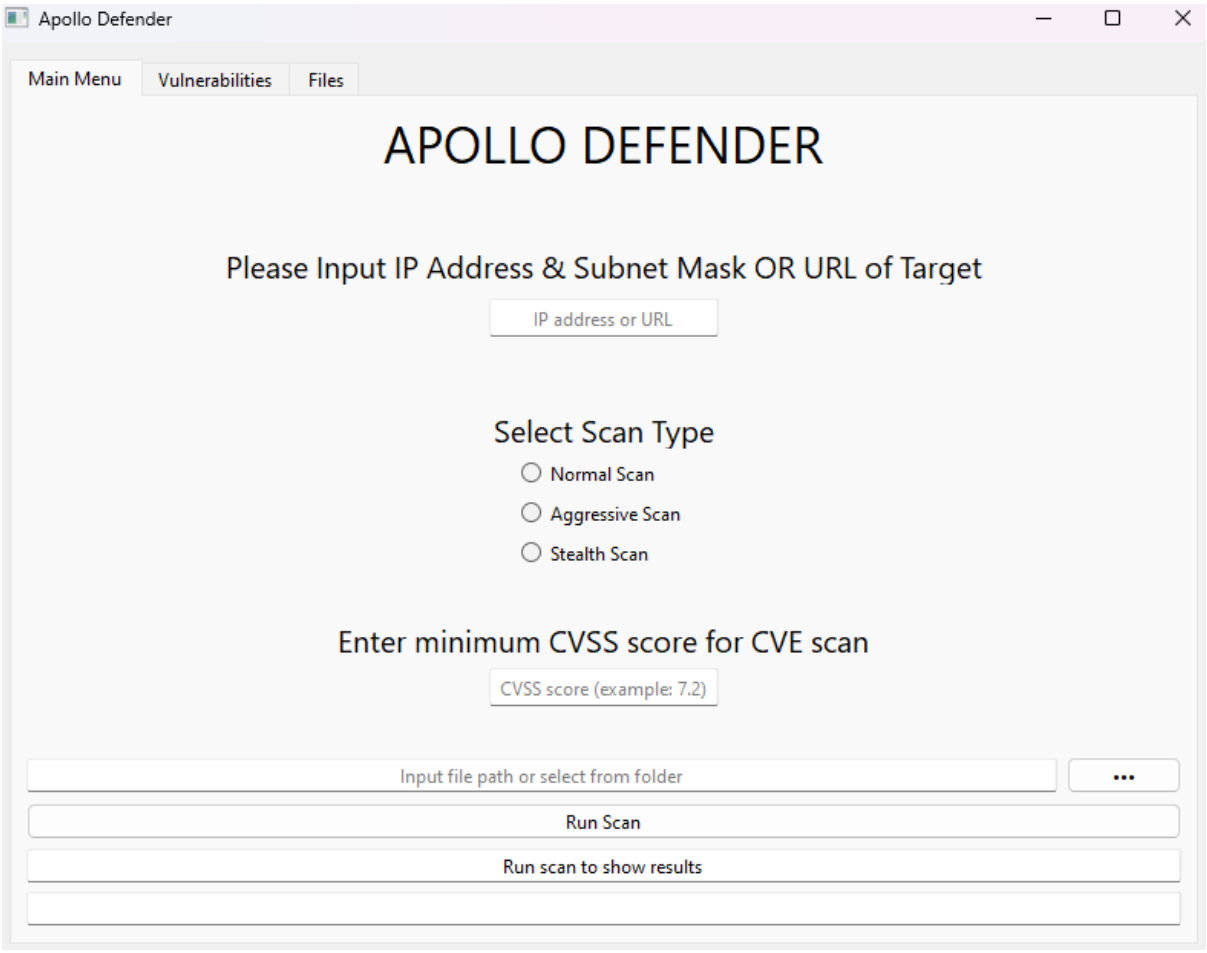


Fig. 2 shows the GUI that I built using PyQt.

The target consumers of this program are people that work at or with an SME to aid in the upkeep of their security in the digital world. For this reason, having a one click button that would detect the network and its IP address was not implemented, as I wanted to allow the user to be able to select a host in their network and run a scan on one particular host should they deem it necessary. Having a button that does this for the user would not allow them to specify a particular target. For these reasons, I believe I have created an interface that is simple to use for people who have a basic understanding of the network that they are dealing with, yet still give them flexibility in both the type of scan that they run as well as the target they choose to scan.

Another thing that I achieved was the successful running of a CVE scan after parsing the information gathered by the Nmap scans. When the program runs, it first completes an Nmap scan of the target network, device, or URL and creates a list of available and active hosts on the scanned network. The list that is returned contains only the IP addresses of these active hosts and then runs a secondary Nmap scan on each of these hosts using a script to detect CVEs, which queries an API to obtain known CVEs.

A third thing that I achieved was the successful parsing of the CVE scans and the subsequent displaying of the CVEs to the user on the UI dynamically. This was an important feature for me to implement as it is the way that the user can easily see all the vulnerabilities on the devices that they have scanned in one central place without having to read through lots of JSON files. *(Fig. 3)*

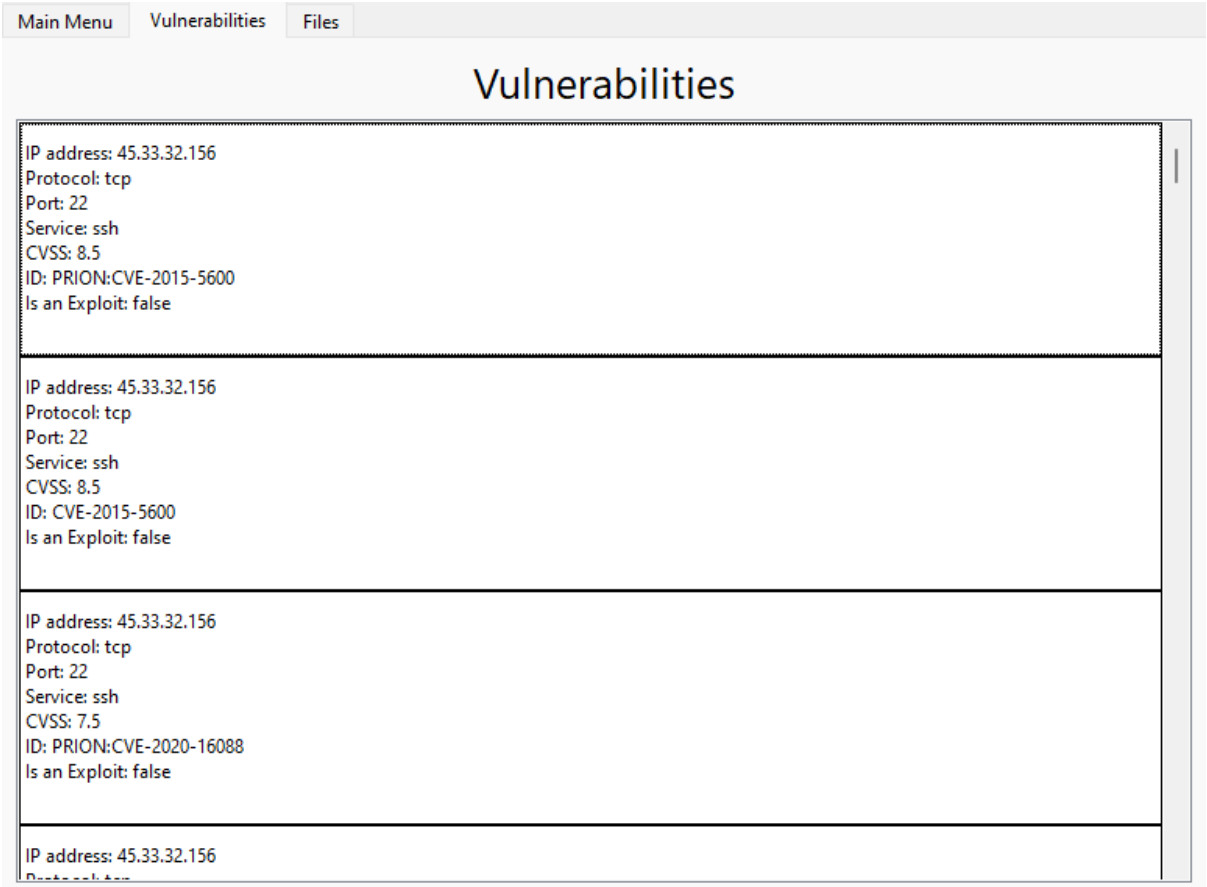


Fig.3 shows the vulnerabilities found after scanning the scanme.nmap.org domain.

What I Didn't Manage to Implement

One of the features that I hoped to implement was sending a report of recently discovered vulnerabilities to users. This feature would have allowed reports and/or alerts to be sent to the user automatically, should the program detect any new vulnerabilities in the network. This would have required the application to be constantly running scans of the network in the background. I did not have enough time to learn how to do this in the time I had available. It also posed the significant challenge of optimising the scans run by the system so that the scans would not cause a notable decrease in network performance or speed. Given more time, this feature could definitely be implemented, increasing the usefulness and effectiveness of the application. The application does still allow for the user to create the reports of the vulnerabilities detected and output it to a file path of their choosing once the scan has been run by the user manually.

The other thing that I did not manage to implement was the function to track the vulnerabilities that have been detected and keep a log of which vulnerabilities have been resolved since the last scan of that target. The basic functionality of this feature has been implemented, like automatically running a CVE scan on the target to pick up all the vulnerabilities, not just ones above the specified CVSS that use user provides. This is something that, given more time, could definitely be implemented and would greatly improve the usefulness of the application. The aim of this feature would be to help improve the compliance of SMEs as it would allow them to accurately keep track of the vulnerabilities found in their network and keep a record of what vulnerabilities have been resolved and when.

A final thing that I did not manage to implement fully was a colourful and styled GUI. The GUI that is available to the user is simple and does allow them to start a scan relatively easily and get a report of the CVE's that have been detected, but I feel like, given more time to develop and learn the finer details of the PyQt framework, I would have been able to implement a nicer looking GUI for the user.

What I Would do Differently if Starting Again

If I was starting this project again, I think one of the main things that I would do differently would be the in-depth research into the different Python libraries that I could use when completing the project. When it comes to Python libraries, I think I severely underestimated the amount of useful Python libraries that would be available for completing the tasks that I needed to complete. There were multiple libraries that I found out about only during my development process that would have been very useful to have found before I started so I could have implemented them sooner. There is also the concept of what you know you know, what you know you don't know, and what you don't know you don't know. I feel like there are plenty of Python libraries that I know of and know how to use as well as libraries that I know about, but don't know how to use, but I have no idea how many libraries that I don't know about. Doing more research would have expanded my knowledge of libraries that I could have potentially used.

Another thing that I would do differently if I was starting again would be my time management. Even though I believe I managed my time well enough for this project, I also believe I could have had more time to implement additional features or complete the ones I have already implemented to a higher standard if I had more clearly defined goals and a more well-planned project timeline.

One of the main things that I would have done if redeveloping this project would be to not set unrealistic expectations for the goal of the project. In the functional Specification and project plan document, I created mock-ups of the GUI which contained features that in hindsight were out of reach for the time of this project. I think when I was creating the functional specifications, I underestimated the scale of the project and the challenges that I was going to face along the way, so I was a lot more ambitious with the features that I was suggesting.

Another thing that I would have done is been more targeted with the time that I spent on the project. When I started, I thought I would be able to spend 12 hours a week on the project. However, this ended up being closer to 8 hours a week. Most weeks, the amount of work I would do in a stint would be around how much work I should have been doing if I was spending 12 hours a week on the project. For this reason, when it came closer to the deadlines of the deliverables, I was rushing to get some of them done up to a standard that I was happy with. This was due to me not targeting my work as well as I could have done during the time that I was spending on the project. If I was to start again, what I would do is do a better analysis of my schedule and figure out how long I could spend on the project and then organise my work accordingly. By doing this, I would have been able to better plan out what parts of the projects I would have done by when.

Finally, I would like to review my GANTT timeline that I created during the functional specification. This is the timeline that I created for this project. *(Fig. 4)*

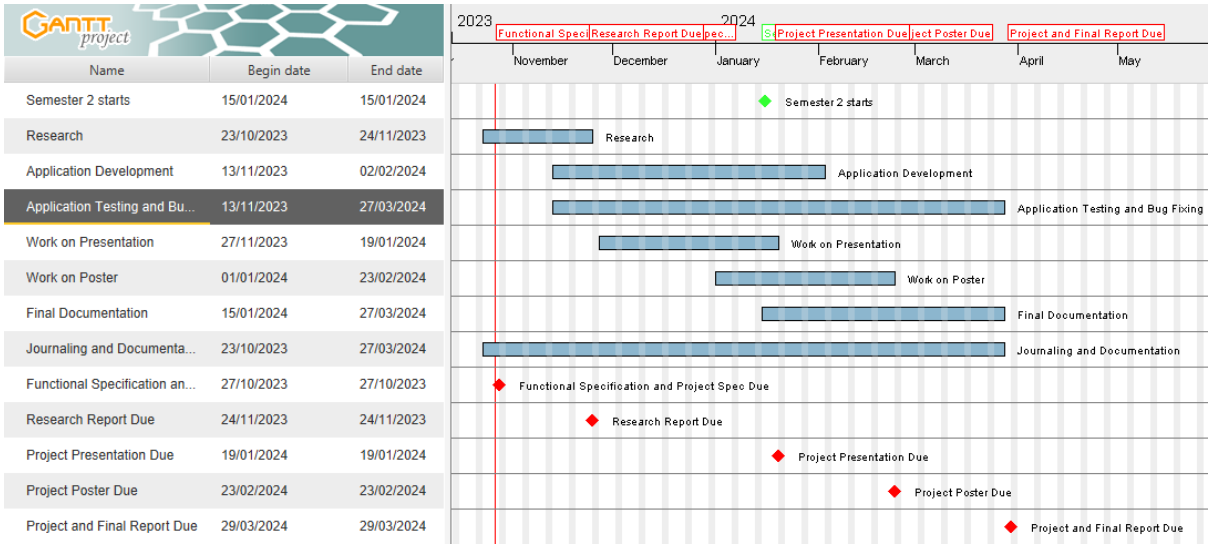


Fig 4. GANTT document of project timeline

As can be seen from the figure above, I had planned to have the development of the application done by the start of February. In hindsight, this was very unachievable. Given the fact that I had many other modules and projects to work on, there would have been no way that I could have got the application completed and fully functional by the end of January. If I was to only work on the project and not work on any other modules during the time period that the timeline is for, then I may have been able to meet the deadlines of this plan. Working on other modules along with running into some difficult issues with the code of the application caused me to quickly fall behind on this schedule. In terms of the documentation that is required for the project, I feel like I mostly stayed on target. The main thing that was off target was the actual development of the application. If I was to start the project again with this knowledge in mind, I would have done a better job of planning the timeline based on all the other responsibilities that I needed to fulfil along with completing the project. By doing this, along with setting aside specific times to work on my project, I feel like I could better meet the deadlines I set for myself.

The Experiences I Gathered

This section will discuss the experience I gained from this project. For me, the main experience would be the presentations.

During the presentations, I got lots of feedback on the approaches that I was taking to developing this application. I got some advice on ideas to look into as well as possible features that would be useful for an application such as Apollo Defender. I took all of these suggestions and questions into account as I continued to develop the application and hopefully these recommendations show in the final product.

Another thing that was a good experience for me was doing the actual presentations themselves. With a time-limit of only 5 minutes, I was challenged to fit all the important information about my project into a short time window. This meant I had to get information across to the audience such as the aim of my project, what my project does and how it intended to use certain technologies to achieve the things I described. The experience of gathering feedback from rehearsals as well as the first run through of my presentation allowed me to further narrow down the information that was needed in order to explain my project well, and I feel like this was a very valuable experience for me. This experience will assist me in the creation of better and more concise presentations in the future as well as help me understand the type of information that is important to an audience when giving a presentation on something like Apollo Defender.

Another piece of experience that I gained from completing this project was that AI, when used correctly, can be a very useful learning tool that reduces the time spent searching the internet for examples of code, and increases the amount of time that can be spent learning about the syntax and control flow of the code. AI significantly accelerated my development time when I used it to better understand the syntax of specific functions and widgets in the PyQt framework. This was because I was able to quickly obtain a code example of the feature I was trying to implement, comprehend that example, and then write code using the knowledge I had gained to implement the desired feature.

One final experience that I gathered while completing my final year project was that of constantly testing my application and improving it or fixing bugs that I found. While I have done this process with smaller programs and applications, I have never done this on such a large scale. The scale to which I was doing this was much greater than anything that I have done before and as a result, it was more of a constant thing throughout the entire application development. Sometimes, when trying to add a new feature to my application, I would end up breaking a feature I had implemented previously, needing me to first go back and fix that issue before continuing with the current task. I needed to do this as a lot of my application code is sequential, meaning that functions deeper in the code require the previous functions to have worked properly. I think that this experience of having to constantly test and debug my application has been beneficial and will help me to develop more robust and secure functions, and therefore applications, in the future.

The Challenges I Faced

One of the main challenges that I faced when staring this project was that of having to learn a new scripting language to complete this with. Since this application relies heavily on using scripts and parsing of large JSON documents, the ideal language for this is Python. This is a language that I had a small bit of experience with before starting this project from messing around with small scripts in Virtual Machines and Capture the Flag style games on websites such as "TryHackMe" and "HackTheBox". Having a strong foundation in multiple other programming languages also helped when starting to fully learn Python. However, even having a solid foundation in other languages did not mean that I was able to start immediately coding fluently in Python. I used some online resources such as Stack Overflow and YouTube channels dedicated to teaching Python and providing tips for Python to help me pick up the basics of syntax and control flow, as well as more complex concepts such as list comprehension which plays a large part in the effective parsing of the JSON files that result from the Nmap scans. In some areas I also used an AI called "Blackbox" in order to help me understand the syntax of more complex parts of the PyQt framework. I would provide the AI with a prompt such as *"I am trying to add an item to a list on my GUI dynamically. What would be the PyQt syntax that I would use to order for me to achieve this"*. The AI would then give me an example of how to add an item to a list widget. I would then study the code that it gave me, understand how and why the syntax worked, and finally adapt it and implement it into my own application. By doing this, I was using the AI as a learning tool similar to reading examples on StackOverflow and gaining knowledge from there, rather than just copying what the AI provided me without understanding it first. Given these challenges, I feel like I have overcome them to the best of my abilities, creating a functional GUI based application that makes effective use of the concepts that I have learned from teaching myself Python.

Another challenge that I faced when starting this project was the idea of creating a visually appealing yet simple GUI for users that would make the application intuitive to use and easy to pick up. I quickly found out that there are limited GUI frameworks that work well with Python. All in all, there are only approximately 4 mainstream GUI frameworks that work well with Python. From my research, I concluded that using the PyQt framework would be the easiest as it somewhat resembles the GUI framework provided natively in Java, a language that I studied for 3 years and have a very good understanding of. There were still many differences once I got into the details of using this GUI framework that all had to be worked around. PyQt is also heavily limited in the customisation that is available for elements, making it difficult for me to create a visually appealing GUI for users. Since the application offers instructions and errors to the user if they input data that would cause the application to run incorrectly the user should find it relatively easy to open the application and understand what is required for a scan to be run, thanks to the straightforward and intuitive GUI that I have created. So, in this way, I feel like I have overcome this challenge to the best of my abilities.

When I returned to college in January, I started multiple new modules, all of which had some form of project or Continuous Assessment implemented into them. This significantly slowed down the development of Apollo Defender as I was required to dedicate large chunks of time to these projects and assessments which subsequently decreased the amount of time that I could spend developing Apollo Defender. On average, I needed to spend 40 hours per week on college related work and attending lectures and labs and of those 40 hours, 12 hours a week should have been dedicated to the project. Due to the amount of time that was required to be spent on other projects and assessments, I wasn't always able to dedicate the full 12 hours to the project every week. This challenged me to efficiently divide my time between all my various modules, projects, and assignments. It additionally encouraged me to optimise the time I spent on each module or project, because any time spent on one took away from the development of Apollo Defender. All things considered, I feel like I have managed my time well between all my projects and assessments and have dedicated as much time as possible to this project without causing myself to burn out.

One of the final challenges that I would like to mention that I faced was the challenge of explaining my project, as well as its goals, technologies, and reasoning in under 5 minutes. The time-limit challenged me to compress my ideas into short, yet simple to understand chunks. I found this a challenge and it required me to do multiple practice runs and make plenty of changes to my presentation in order to keep it within the 5-minute time-limit. This wasn't easy for me as I felt like there was plenty of information that was in my presentation that would be important for the listener to hear in order for them to fully understand the concept and aim of my project. I felt that after a lot of preparation and editing, the presentation would offer the audience a decent picture of what my project was about while also allowing me to provide details and specific points should there have been time for questions following the presentation.

How I've Grown Because of this Project

There have been many ways that I have grown as a person because of the time I have spent completing this project. Some of the ways that I have grown include improving my problem-solving skills, improving my technical proficiencies, realising my ability to adapt to and overcome challenges, and an overall increase in my confidence when it comes to completing large projects and assignments.

The first thing that I would like to discuss is the improvement of my problem-solving skills. I came across many issues during the development of my project that I did not expect to encounter. However, I was able to take breaks when necessary and come back to the code with a fresh set of eyes, which helped me have a different approach and outlook to a lot of the problems. With a bit more of a targeted approach to the problem and the potential solutions, this break and fresh viewpoint nearly always helped me overcome the problems I was facing at that time. Although I think I've always been quite proficient at this, I definitely got better at it while working on this project, because there were some tough challenges that I had to overcome at different stages of the development (e.g. parsing all the JSON files) that needed to be fully understood and solved using multiple approaches.

The second thing that I would like to mention is my improvements in my technical abilities when it comes to using Python especially. Python is at the core of my project and as a result, I was doing all my project in the one coding language, meaning that I got a lot of experience using it. This time spent solely using Python to code my project has been a great experience as it gave me the chance to fully immerse myself in Python and learn as much about it as I could in the given time. Since I started my project, I feel that my technical abilities when it comes to writing and utilising the vast array of powerful features that Python offers have improved greatly. I have become comfortable with the basic syntax of Python as well as the syntax of more complex structures. I feel like this will be a great benefit to me further on in my career, as Python is a language that is heavily used in the cybersecurity industry .

The third thing that I would like to reflect on is my time management and the improvements I have made to it over the course of this project. When I started this project, I felt like my time management was good overall, but as I started getting further into the timeline of the project, I realised there have been a few areas where I could have managed my time better. One thing that I failed to do while completing this project was allocate specific times that I could use to work on it. This, combined with the fact that I lost track of how much time I was able to devote to the project each week, meant that not all of the project's components were finished to the best standard I am capable of. I do still feel that the time that I spent working on the project was targeted pretty well. As with any project that is being reviewed in hindsight, there were areas that I could have improved in and targeted my work better but given the challenges that I faced and the time that I had, I think I completed my work to a high standard. I believe that the time

management experience I had while finishing my project helped me learn a lot about how to better manage my time going forward and how to make better preparations and time schedules to meet all of the project deadlines.

The final thing that I would like to discuss is the way that I have improved my adaptability, resilience, and work ethic when it comes to facing challenges. Throughout the project, I came across many challenges in terms of time management, nervousness about presenting my project, or logical challenges in relation to the code that I was writing. When it came to giving the presentation on my project to the class, I was quite nervous even though I had rehearsed the presentation multiple times before the day. In the end, the presentation went well for me, and it gave me the confidence to improve my presentation and be more proud and confident of my work in the second presentation. I felt like the first run-through of the presentation helped me build some resilience and help me believe in myself for any future presentations. The main thing, however, was to do with issues I faced when writing my code. When I was coding my project, not only was I learning an entirely new coding language as I was going, but I was also trying to figure out problems with the logic that I had implemented and also figure out why errors were occurring. I think I learned something about myself when it comes to this aspect. I found out that I am quite adaptable and resilient when it comes to doing my coding. There were plenty of times where I came across a difficult or confusing problem while coding that, at first glance, seemed daunting. However, I kept working diligently on the problem, often for hours on end until I had resolved it. Throughout the time that I spent trying to fix these errors, I very rarely lost the motivation to keep working on it, and I believe that it is this drive that helped me get my project to the stage that it is at. This time not only helped me realise this about myself, but also helped me build and develop it further. From doing this over and over again, I learned to not get worked up about a possible solution not working, but to rather regroup and assess why that possible solution didn't work, and then further improve it in the hopes that the new version would solve the problem that I was having at the time.

Glossary

SME	->	Small to Medium sized Enterprise
JSON	->	JavaScript Object Notation
GUI	->	Graphical User Interface
NVD	->	National Vulnerability Database
API	->	Application Programmer Interface
CVE	->	Common Vulnerabilities and Exposures
CVSS	->	Common Vulnerabilities Severity Score

Declaration of Plagiarism

I declare this document in this submission in its entirety is my own work except for where duty acknowledged. I have cited the sources of all the quotations, paragraphs, summaries of information, tables, diagrams, or other material. This includes software and other electronic media in which intellectual property rights may reside. I am aware that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offense.