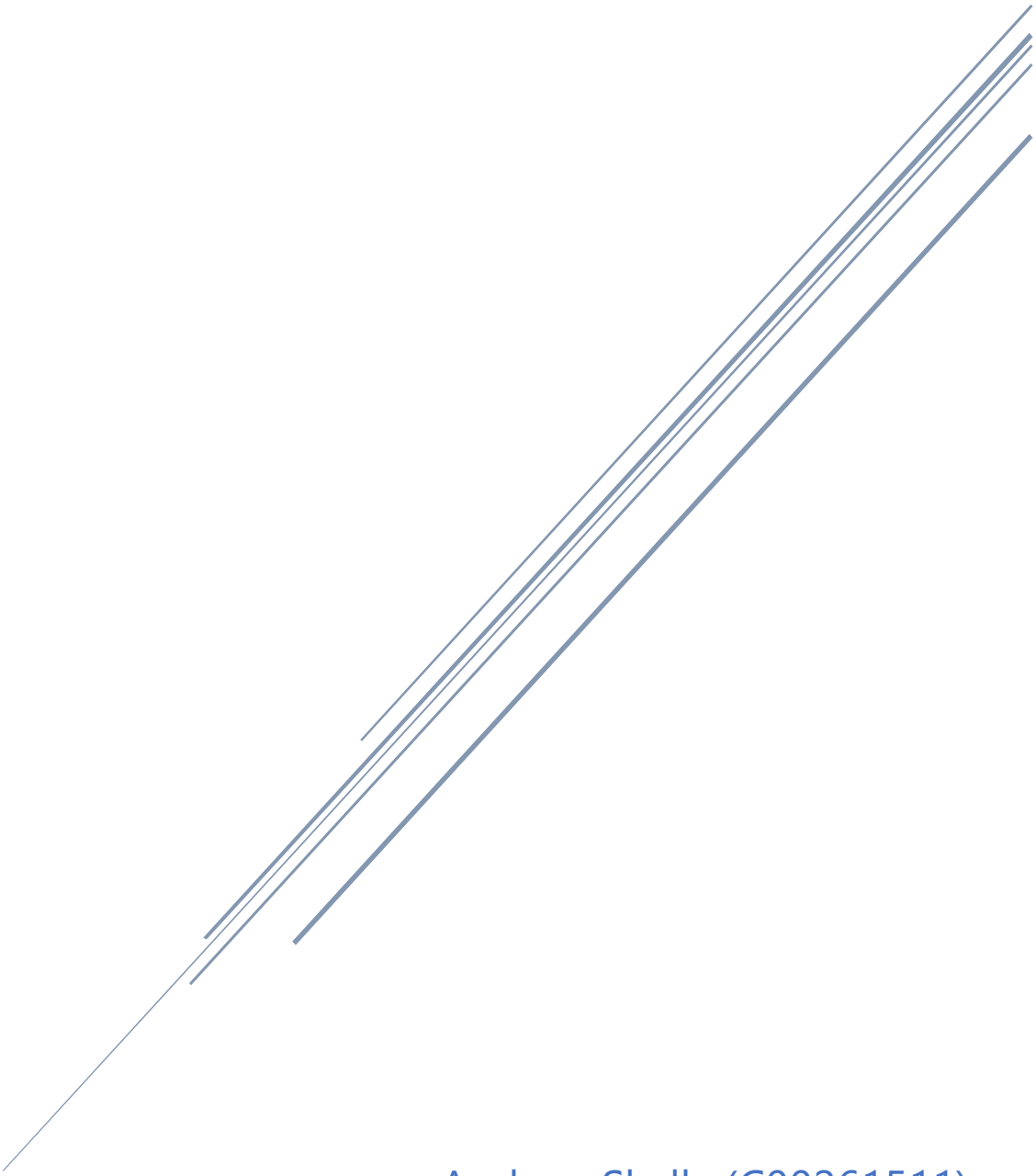
ASSET IDENTIFICATION AND RISK ASSESSMENT FOR SME'S

Functional Specification and Project Plan



Andrew Skelly (C00261511)
Cybercrime and IT Security

Abstract

Asset identification and risk analysis are essential actions for SMEs to carry out in the modern world. It will help an SME improve their overall security and also save time. This Functional Specification and Project Plan will detail a plan for a GUI based Python network scanning and vulnerability identification tool to help SMEs efficiently and effectively identify their IT based assets and understand the vulnerabilities that their assets may be at risk from. This application will leverage widely used cybersecurity technologies and information such as Nmap and NIST's database of categorised Common Vulnerabilities and Exposures (CVEs) (*NVD - CVEs and the NVD Process*, no date) to detect an SMEs devices and help them understand their network and the vulnerabilities it may face in a simple and easy to read format.

This application will be called Apollo Defender.

Table of Contents

Introduction

Overview

Cyberattacks on SMEs are increasing. Whereas large organisations are able to afford sizeable cybersecurity teams, SMEs, especially smaller ones, cannot. I will develop a toolkit, aimed at small companies, that takes them through the process of identifying their assets, and makes it easy to understand risk assessments and recommendations.

Full Description

In the current cybercrime risk assessment and threat avoidance landscape, there is a clear "lack of simplified and practical risk management solutions aimed at the SME sector" ,as identified in their report "Cybersecurity for SMEs: Challenges and Recommendations"(*Cybersecurity for SMEs - Challenges and Recommendations — ENISA*, 2021). This project aims to fill that gap and provide SME's a simple, yet practical toolkit to guide the business through the identification of their assets, as well as the identification of possible risks to these identified assets. The toolkit will also help make appropriate recommendations to the SME based on the risk tolerance willing to be accepted and the severity of damage caused to the SME should any of the identified assets be compromised.

This toolkit will be built from the ground up with security in mind. Due to the sensitive nature of the information that will be captured by the toolkit to help identify possible vulnerabilities, security of this information will be of the utmost importance. A toolkit like this could easily be used by threat actors as a reconnaissance tool and as such, it must be ensured that the capabilities of this toolkit are limited in such a way as to minimise the risk of threat actors using it as an effective network enumeration tool.

Scope

The scope of this document is to detail the plan and explain the first ideas for Apollo Defender as well as setting out a timeline for the completion of each critical part of the project and application. In regard to the scope of the application itself, it will be a GUI based Python application that will allow network administrators to define the scope of their scans and gather valuable information about their network and subsequently provide them with a list of CVE's that devices on the network may be vulnerable to. Apollo Defender will not attempt to exploit any vulnerabilities.

The Details

This tool will use state-of-the-art asset identification to help SME's identify risks to their assets. This will be in the form of network messages sent out to all devices in the environment. These messages will then return information to the toolkit such as device type, operating system type and version, available and open ports, and versions of software (e.g., Apache) that a machine might be running.

Then, based on the information received from each device, the toolkit will then cross reference vulnerability listing from NIST to hopefully identify any possible vulnerabilities that hosts, or other network devices, may be susceptible to. This process will be repeated for each device on the network and will then be categorised for easier viewing of the results.

In cybersecurity, when talking about assets, there is a certain level of risk that a business or individual can accept before the risk mitigations and protections outweigh the value of the asset itself. This is in part down to what the asset is used for. For example, a vital database will be more important to a business than an employee's laptop so it will require more protections. Based on the value of the asset physically and digitally(the data stored on it), an individual or business can conclude on an acceptable level of risk for that particular asset. This would usually involve placing less protection on a less valuable device or a device that stores less valuable data as the cost of protecting it would be more than that of risking losing the data or device. Once the application's preliminary scan is complete, the tool will then prompt the user to input their acceptable level of risk and will allow the user to select a severity for each vulnerability. This severity should be determined by the owner or other trusted party of the company and should relate to how much damage would be caused to the business should the identified assets be compromised.

Finally, the tool will then give recommendations based on the vulnerabilities identified and the impact severity should the vulnerable devices be compromised. The user can then use these recommendations to make changes to their environment so that it is protected from possible attacks. Apollo Defender will also generate a report for system administrators which will give them a way to show compliance by proving that they mitigated CVEs and other threats in a reasonable period of time.

Deliverables

This section will detail the deliverables that are required for this project's completion.

- Functional Specification and Project Plan
- Research Poster
- Research Report
- Presentation
- Final Report
- Functional Application
- Downloader / Installer
- Website

Technologies

Language

Python 3.12- Due to the benefits of python such as being an ideal scripting language that is widely used in the cyber security area, I will be using Python to program my application. I also believe that the relatively simple nature of the language itself, it will make it easier for me to learn this programming language, as I do not yet know Python fluently.

PyQt – after doing research on Python GUI frameworks that would suit my needs best, I have decided to use PyQt. This is because it is relatively simple, well documented GUI framework. It is slightly more advanced that the GUI framework that comes packaged with Python, but it will allow me a greater deal of customisation with the different GUI elements.

Tools

Nmap – I will be using Nmap for the network scanning element of my application. It is a trusted network scanning tool that is used in many settings, and it will provide me with a large amount of information that I can use to efficiently categorise an SME's assets.

NVD API – The application will use the NVD API as a way to keep up to date on the latest CVEs that are released and update its knowledge, therefore keeping a network more secure with regular updates to the list of CVEs that it would be scanning.

Virtual Network spoofing tool – During my time at Trend Micro for my third-year internship, I came into contact with people who can host virtual networks, so I will be using this as a way to create a virtual network to test and showcase my application in a secure and safe environment.

Database

MySQL – I will be using MySQL to create databases for each SME and securely store their device information. This will help in making the identification process of devices more efficient as the tool will already have a database of known devices on the network and will not need to reidentify all devices.

Platform

Due to the fact that Apollo Defender will be developed using Python and any libraries required will be packaged with Apollo Defender, this system will be capable of running on both Windows and Linux operating systems. It will be GUI based and developed using the PyQt framework. This framework will allow me to organise and format GUI elements.

FURPS

Functionality, Usability, Reliability, Performance and Supportability (FURPS) is a method of classifying software quality attributes first developed by Hewlett-Packard(*FURPS - Wikipedia*, 2022).

Functionality

Process

Apollo Defender will go through 4 core steps when performing its actions.

1. Network identification: At this stage, a network administrator will provide an IP address and subnet mask for the portion of the network they will be scanning.

2. Scanning: Next, the application will feed the IP address and subnet mask provided to Nmap, which will run an intensive of light scan, depending on the networking administrator's needs. This information will be saved to a database.

3. Vulnerability checking: The application will interface with the NIST database of CVE's and identify any vulnerabilities that the devices on the network may be vulnerable to.

4. Report generation: Finally, the application will generate a report that can be saved by the network administrator for review. This will also include statistics such as vulnerability generation time (i.e., The time that the toolkit detected the vulnerability) and mitigation times. This will allow the SME to prove that they resolved a vulnerability in short order.

Features and Security

Some of the main features that will be showcased in this application will include the following:

- Securely stores login credentials.
- Allows system admins to specify the range of their search. A whole network scan will be completed if no range specified.
- Allow system admins to view CVEs that are exploitable in their networks.
- Generate a report for system admins, which will show assets on the network, CVEs detected in the network, the time these CVEs were detected and the time these CVEs were mitigated.

Some of the security features of Apollo Defender will include:

- Multi-factor authentication for login
- Secure encrypted connection between the application and server
- All sensitive data stored in database will be securely encrypted using SHA3-256
- Each copy of the application will be locked to a specific machine using it's MAC address, ensuring that copies of the application cannot be distributed easily.
- The user will automatically be logged out after 10 minutes to ensure that security is upheld.

Usability

Misuse Case Diagram

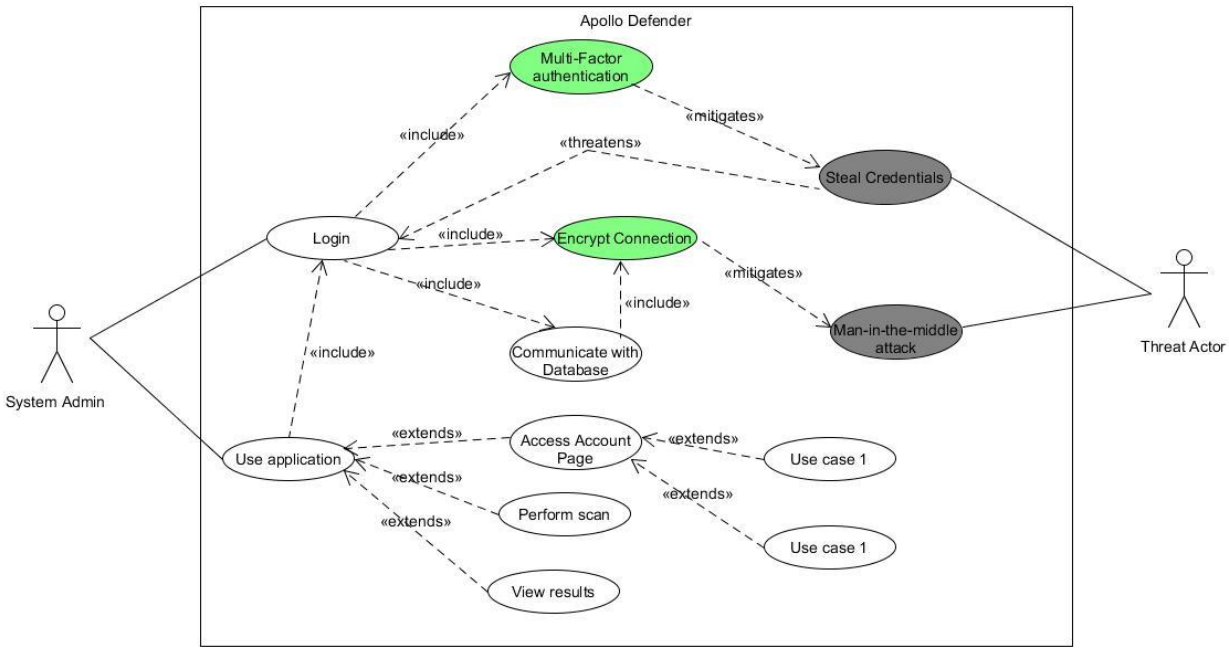


Figure 1: Misuse Case Diagram for Apollo Defender

The usability of Apollo Defender will be aimed towards system administrators of SME networks. The application will have a simple and intuitive interface making it easy for system administrators to quickly and efficient operate scans. For the system administrator to gain access to Apollo Defender, they will first have to go through an authentication process which will ensure that they are a system administrator and ensure that the application is locked to their MAC address.

Reliability

Apollo Defender will mainly rely on the connection between the customer and the NIST NVD API. This will ensure that customers' databases are frequently updated with the latest CVEs. Another reliability for the application will be the central server that they will authenticate against when logging in. If this server goes down, then no user will be able to log in, therefore causing downtime. This will be mitigated through the use of backup servers that will kick in should the main server go down.

Performance

As this tool will perform a full network scan when it is first used, it may have a high impact on the performance of the system on which it is running. The length of the scan will depend on the size of the network. After this first scan and setup is complete, however, the application will become much more efficient and only have to be run when new devices are connected to the network to add them to it's knowledgebase. This will decrease scan times significantly.

Supportability

Apollo Defender will be fully packaged, so system admins will not be required to install any other programs or services before installing Apollo Defender. The application will have Python packaged and will install Python on the system if it detects that it is not present. In this regard, there should be no issue with supportability as Apollo Defender will be capable of running on any Windows and Linux machines that is capable of running the latest version of Python.

GUI Example

These two images showcase the log in and main page general layout. This is a very basic layout and will more than likely evolve over time, but these images hope to reflect the final product.

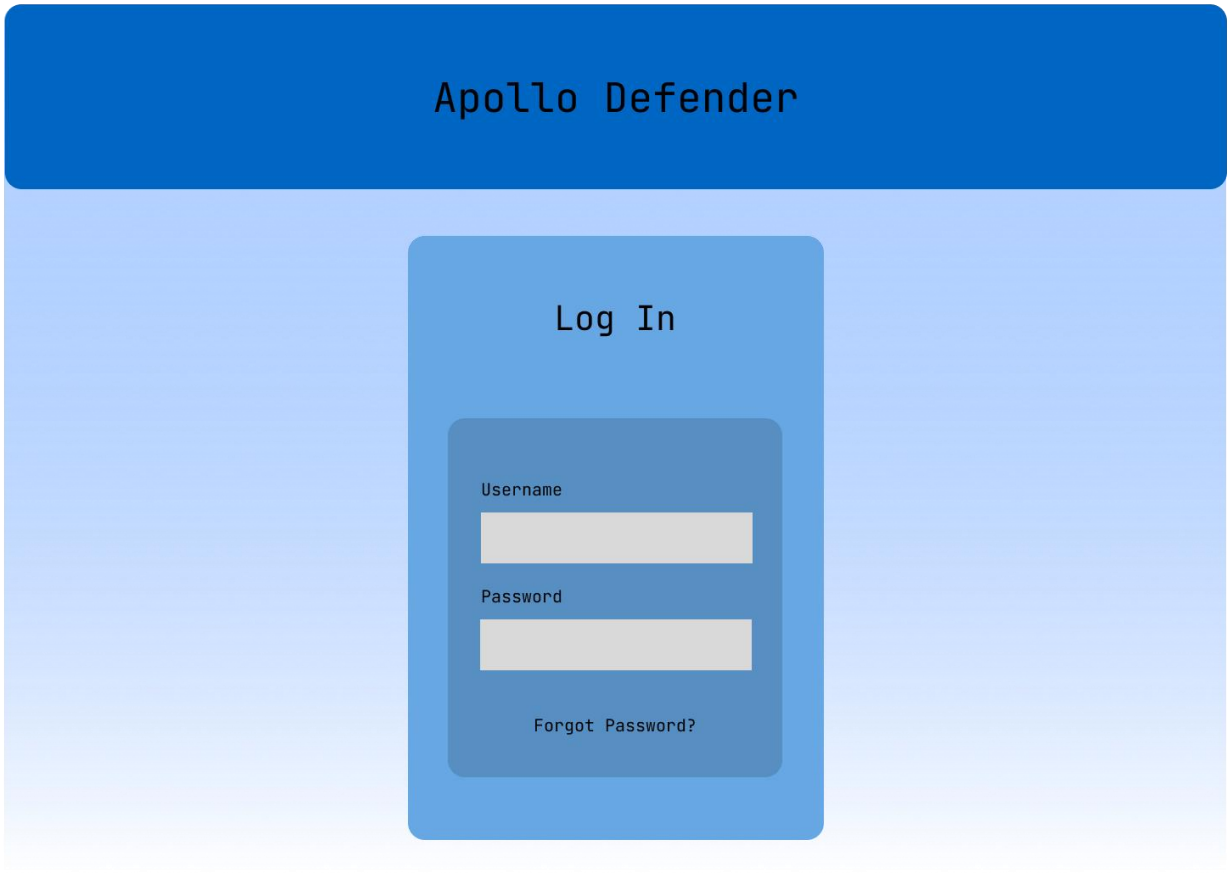


Figure 2.1: Main Log-In Page

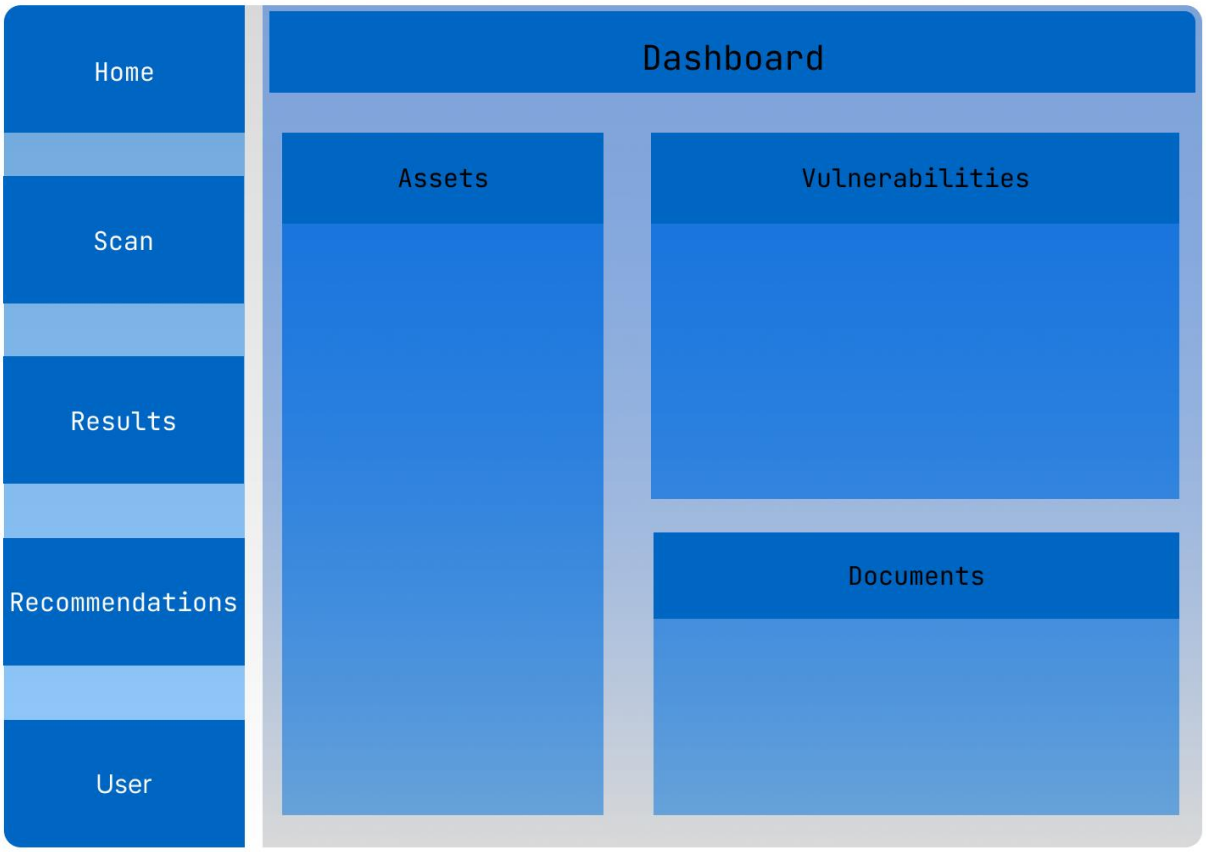


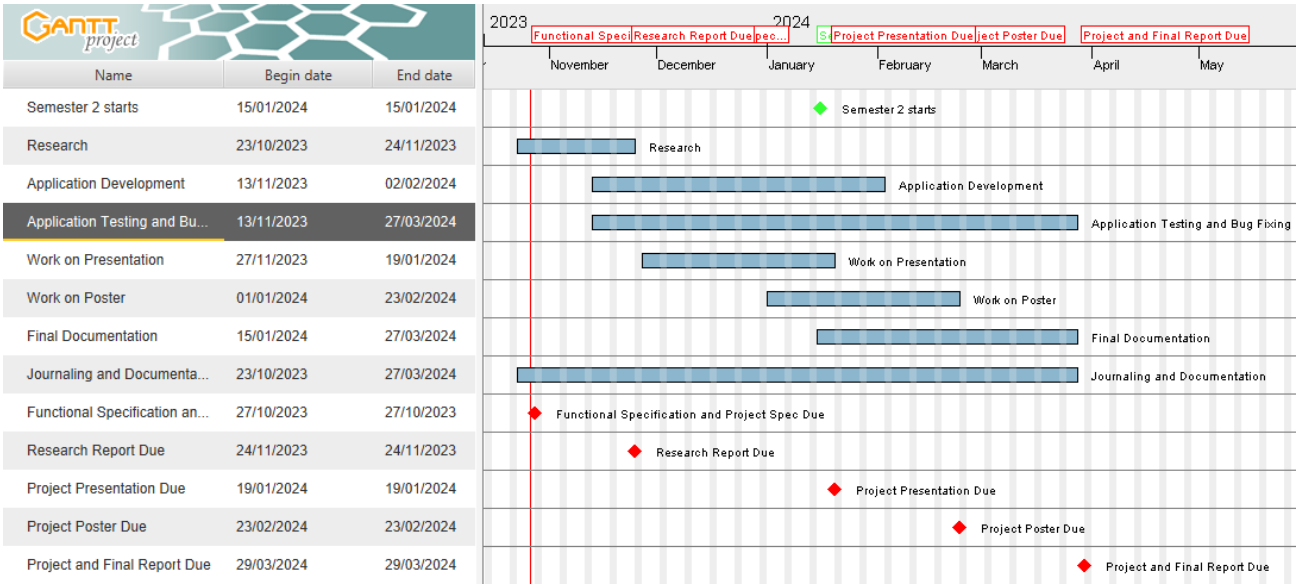
Figure 2.1: Dashboard Main Page of Application

Project Plan

The Plan

I plan to develop Apollo Defender from the very start with a security by design approach. I intend for this application to be fully packaged and be able to run on both Windows and Linux operating systems. In regard to security by design, the application should have a log in that ensures that not only are the login details of the person correct, but they are also accessing Apollo Defender from a verified device, identified using the devices' MAC address.

GANTT Chart of Project Timeline



Name	Begin date	End date
Semester 2 starts	15/01/2024	15/01/2024
Research	23/10/2023	24/11/2023
Application Development	13/11/2023	02/02/2024
Application Testing and Bu...	13/11/2023	27/03/2024
Work on Presentation	27/11/2023	19/01/2024
Work on Poster	01/01/2024	23/02/2024
Final Documentation	15/01/2024	27/03/2024
Journaling and Documenta...	23/10/2023	27/03/2024
Functional Specification an...	27/10/2023	27/10/2023
Research Report Due	24/11/2023	24/11/2023
Project Presentation Due	19/01/2024	19/01/2024
Project Poster Due	23/02/2024	23/02/2024
Project and Final Report Due	29/03/2024	29/03/2024

Bibliography

Cybersecurity for SMEs - Challenges and Recommendations — ENISA (2021). Available at: https://www.enisa.europa.eu/publications/enisa-report-cybersecurity-for-smes (Accessed: 27 October 2023).

FURPS - Wikipedia (2022). Available at: https://en.wikipedia.org/wiki/FURPS (Accessed: 27 October 2023).

NVD - CVEs and the NVD Process (no date). Available at: https://nvd.nist.gov/general/cve-process (Accessed: 27 October 2023).