**PhishNet Project Plan**

**Document Version 2.0**

Updated on 11/25/2024

Written By:

Julia Pallazola

Aarshia Hukmani

Gianmarco Bonventre

AJ Whitman

1. **Introduction**

**1.1 Purpose & Objectives**

This document aims to define the strategy our team will follow to develop and complete PhishNet, a malware detection tool built for Ubuntu 20.04. This project plan will detail the task breakdown, each team member's responsibilities, and the completion timeline, ensuring that we meet all project milestones before the final presentation in early December 2024. The main objectives of PhishNet are:

* To create a reliable malware detection and removal tool utilizing signature bytecode detection and pattern matching.
* To provide an open-source and free application exclusively for Ubuntu 20.04 users by December 2024.
* To implement key features such as scheduled scanning, on-demand scanning, real-time malware detection, and quarantine or deletion of infected files.

**1.2 Scope**

PhishNet is a malware detection and removal application developed specifically for the Ubuntu 20.04 operating system. The software will leverage ClamAV’s open-source virus definitions and signature detection to scan files, folders, and entire drives for malware. Users will be able to perform scheduled scans (daily, weekly, or monthly), select specific files or directories for scanning, and update antivirus definitions regularly to maintain up-to-date protection. PhishNet will quarantine or delete infected files based on user preferences and maintain a history of quarantined files for future review. The scope of PhishNet is limited to malware detection and removal; it does not handle phishing threats or advanced cyberattacks. Additionally, its effectiveness is dependent on timely updates from the ClamAV virus definition database.

**1.3 Definitions, Acronyms, and Abbreviations**

| **Word, Acronym, or Abbreviation** | **Definition** |
| --- | --- |
| GitRepo | A git repository (gitrepo) is a storage location where Git tracks and manages the versions of files and directories in a project. It allows users to save snapshots of their work, collaborate with others, and roll back to previous versions if needed. Each git repository contains a complete history of changes, along with tools to manage and track these changes efficiently |
| SDK | An SDK (Software Development Kit) is a collection of tools, libraries, and documentation that helps developers create software applications for specific platforms or systems. |
| PhishNet | A lightweight antivirus/anti-malware tool, available on Ubuntu 20.04. With a multi-functional scanning tool and an easily accessible history of scan results. |
| GUI | Graphical User Interface |

**1.4 References**

PhishNet GitHub repo:

<https://github.com/aarshiah/PhishNet>

ClamAV GitHub repo:

<https://github.com/Cisco-Talos/clamav>

ClamAV documentation:

<https://docs.clamav.net/>

PhishNet Software Requirements Specification:

<https://github.com/aarshiah/PhishNet/blob/main/Requirements%201.0.pdf>

1. **Project Organization & Overview**

**2.1 Team Info & Roles**

| **Name** | **GitHub Profile** | **E-Mail Address** |
| --- | --- | --- |
| Aarshia Hukmani | https://github.com/aarshiah | ahukmani1@pride.hofstra.edu |
| Gianmarco Bonventre | https://github.com/Gianmarcob7 | gbonventre1@pride.hofstra.edu |
| Julia Pallazola | https://github.com/jpallazola | jpallazola1@pride.hofstra.edu |
| AJ Whitman | https://github.com/Barrelya | awhitman1@pride.hofstra.edu |

**2.2 Project Deliverables**

The PhishNet development team will deliver a fully functional malware detection and removal tool for Ubuntu 20.04 by early December 2024. The software will be capable of detecting malware using ClamAV’s signature bytecode detection and pattern matching. The final project presentation will be given on the week of December 3rd, 2024. The master branch of the PhishNet GitHub repository will contain the final working build by this date. All deliverables will align with the software requirements in the software requirements specification document, ensuring PhishNet meets its core functionality and user requirements.

**2.3 Project Milestones**Milestone 1: Integration of Clam AV as the core backend for PhishNet’s scanning engine and frontend framework for the PhishNet user interface (Tasks I-1, I-2, R-3). The estimated completion date is October 10th, 2024.

Milestone 2: Completion of the GUI design, scanning, and schedule page (Tasks G-1, G-2, G-3). This includes the basic functionality of initiation scans. The estimated completion date is October 20th, 2024.

Milestone 3: Implement the malware scanning functionality (Tasks S-1, S-2, S-3). The software will support Quick Scan, System Scan, and Custom Scan, with backend integration using ClamAV. The estimated completion date is October 30th, 2024.

Milestone 4: Completion of scan scheduling and reporting features (Tasks S-5, S-6, P-6). PhishNet will allow users to schedule scans, view history, and receive detailed reports. The estimated completion date is November 5, 2024.

Milestone 5: Full integration of quarantining, database update capabilities, and scan reports (Tasks P-4, A-1, P-6). The estimated completion date is November 20th, 2024.

Milestone 6: Final testing and deployment (all high-priority tasks). The code will be reviewed, tested for bugs, and optimized for performance by the team. The estimated completion date is November 25th, 2024.

Refer to sections 5 and 6 for more detailed information on task scheduling.

**2.4 Branching Policies**

Branching policies are essential to maintaining a smooth workflow in PhishNet's development process. The main branch will be the release branch, with all completed and approved builds merged here. Below the master branch is the development branch, which will be the main branch for integrating new features and testing. Feature branches will be denoted by “feature/task-id” and will be branched off from development. They will require code reviews and the entire team's approval before merging into development. Similarly, bug-fixing branches, with the prefix “bug/”, will merge from development to master and require the entire team's approval.

**2.5 Project Management & Control**

Team meetings will be held weekly over Discord and scheduled when everyone is available. These meetings aim to assess the project’s progress, address any issues, and plan tasks for the week. Updates will also be done through Discord channels.

1. **Risk Analysis**

**3.1 Project Risks**

The core project risks are implementing ClamAV in the backend, creating the GUI, and the API that translates the GUI events into the backend ClamAV commands. The libclamav library is now old and has little documentation, after an initial learning and setting up period project progress will increase.

**3.2 Product Risks**

The main project risk is ClamAV performing poorly in comparison to other antivirus solutions in terms of detecting malware and production speed. ClamAV is the best free open-sourced virus scanner available, as utilizing the best-fitting software is incredibly important. Phishnet is under the GNU General Public License 3.0. Despite being free software, Phishnet still has business risks. The software is under liability if there are false positives or missed malicious files. Though great caution is taken, software with no errors is not realistic.

1. **Hardware and Software Requirements**

The following section outlines the hardware and support software required to develop the PhishNet application. Failure to adhere to these requirements may disrupt and delay the development process, delaying the estimated schedule.

**4.1 Hardware Requirements**

To test the software, developers will need Ubuntu 20.04 or a virtual machine running a Ubuntu 20.04 environment. ClamAV development is also available for GNU/Linux and UNIX operating systems with the same memory requirements as the Ubuntu version. However, PhishNet will not be able to run in these environments and will require a Ubuntu 20.04 operating system to function properly. Only certain versions are recommended for developing the ClamAV Antivirus with other operating systems. Recommended versions for development can be found on the [ClamAV Documentation](https://docs.clamav.net/Introduction.html) website.

**4.2 Software Requirements**

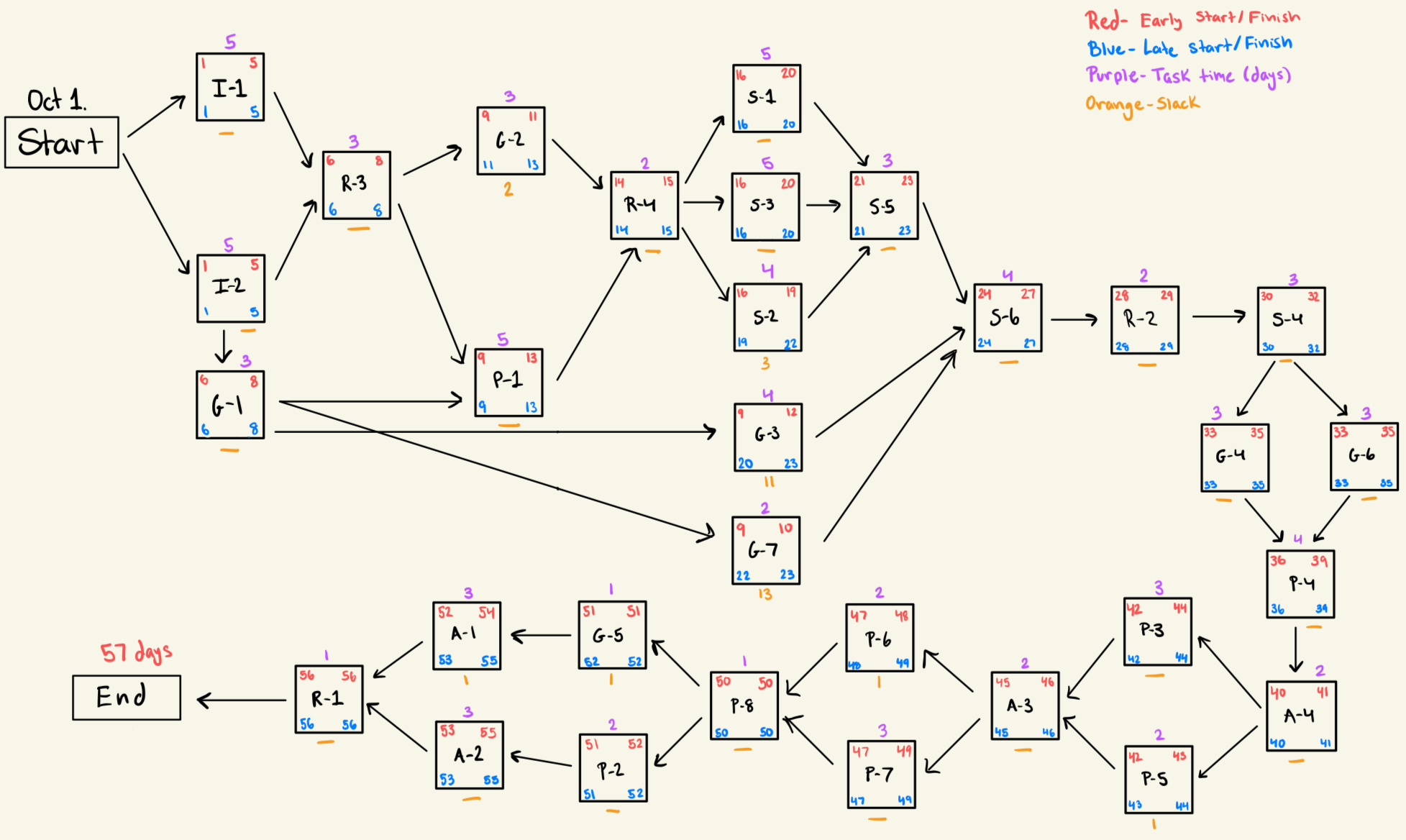
Visual Studio 2022 will be required for the development of PhishNet. Additional requirements include build tools for C++ in Visual Studio 2022, the QT Creator library/dependencies, and ClamAV requirements to run the service (CMake, Rust). This set of requirements will also be needed if any updates or changes are to be made to PhishNet.

1. **Task Information & Scheduling**

**5.1 Task Index**

| **Priority** | **Task Name** | **Description** | **Time to Complete (days)** | **Assignee(s)** |
| --- | --- | --- | --- | --- |
| **High** | S-1 Quick Scan | Scan the user-designated main directory that runs the operating system and essential files to the machine as well as files that have at one point been quarantined | **5** | AJ  Gianmarco |
| **High** | S-2 System Scan | Scan main disk and full file directory of the machine | **4** | AJ  Gianmarco |
| **High** | S-3 Custom Scan | Custom will allow the user to determine what disks, directories, or individual files to scan | **5** | AJ  Gianmarco |
| **Medium** | S-4 Malicious Content Notification | Each scanning method will alert the user of the location of all files found to be malicious | **3** | Aarshia  Julia |
| **High** | S-5 Scan Scheduler | Use the Ubuntu clock to allow the user to choose an interval for scheduled scanning from the options of daily, weekly, or monthly. | **3** | AJ |
| **High** | S-6 Scheduler Scan Type | Scheduled scan will be configured to one of three scan types listed in the scanning methods, with Quick being the default | **4** | Gianmarco  AJ  Julia |
| **High** | P-1 Bytecode Database Cross-Search | Use the ClamAV virus bytecode database to scan files before being downloaded by cross-searching the file with the database | **5** | Gianmarco |
| **Medium** | P-2 Download Protection | When the file is deemed safe it will allow the download, and when the file is found to be malicious the download will be halted and the user notified | **2** | AJ  Gianmarco |
| **Medium** | P-3 Quarantined Files Notification | When an existing file on the device is deemed to be malicious through scan, PhishNet alerts the user | **3** | Aarshia  Julia |
| **High** | P-4 Encrypted Quarantine | PhishNet will quarantine the file by first encrypting the file | **4** | AJ |
| **Low** | P-5 Remove Quarantined File(s) | User will have the ability to remove the quarantined files through the quarantine manager button on the UI | **2** | Gianmarco  Aarshia  Julia |
| **High** | P-6 Scan Report | After a scheduled or User initiated scan, PhishNet will provide a report on all malicious files as well as the location where they were found | **2** | Aarshia |
| **High** | P-7 History Report | Provide a Threat History report, which will allow the user to see which directories in the past have had what malicious files | **3** | Aarshia  Gianmarco  AJ |
| **Medium** | P-8 Flag Files | Files with a history of being quarantined for malware will be flagged and scanned during Quick Scan | **1** | Gianmarco |
| **High** | A-1 Automatic Update Initiation | PhishNet will check for updates needed to the virus bytecode database and GUI (if changes are made) before proceeding with any scans to provide the most accurate protection, if there is no internet connection this step will be skipped | **3** | Julia |
| **Medium** | A-2 Manual Update Initiation | Users can manually check for updates through settings and initiate updates if they are available, if there is no internet connection, it will not be possible to perform an update | **3** | Gianmarco |
| **High** | A-3 History & Report Storage | Through its own program file directory, PhishNet will store Threat History as well as Reports | **2** | Aarshia |
| **High** | A-4 Quarantined File Storage | PhishNet will store Quarantined files through a Quarantined file directory in which no files can be executed | **2** | Gianmarco  AJ |
| **High** | I-1 ClamAV Initialization | Install the ClamAV library for the backend and get an initial run | **5** | All |
| **High** | I-2 Front-End Initialization | Setup the front-end framework and get an initial run | **5** | Aarshia  Julia |
| **Medium** | G-1 Home Page | Create Home Page GUI | **3** | Aarshia  Julia |
| **Medium** | G-2 Scans Page | Create Scans Page GUI | **3** | Aarshia  Julia |
| **Medium** | G-3 Scheduler Page | Create Scheduled Scans Page GUI | **4** | Aarshia  Julia |
| **Medium** | G-4 History Page | Create History Page GUI | **3** | Aarshia  Julia |
| **Low** | G-5 Settings Page | Create Settings Page GUI | **1** | Aarshia  Julia |
| **Medium** | G-6 Last Scan Page | Create a Last Scan Page GUI | **3** | Aarshia  Julia |
| **Low** | G-7 Notifications | Create a Notification pop-up GUI | **2** | Aarshia |
| **Medium** | R-1 Crash Protection | The software will resume the last known state without data loss during failure and computer crashing | **1** | All |
| **High** | R-2 Scanned Device Performance | Run scheduled and real-time scans in the background without causing performance degradation or requiring system restarts | **2** | All |
| **High** | R-3 Application Start | Will start when the system starts | **3** | AJ  Gianmarco |
| **High** | R-4 Files to not be Scanned | PhishNet will not delete or quarantine any files essential to the system running properly | **2** | AJ  Gianmarco |

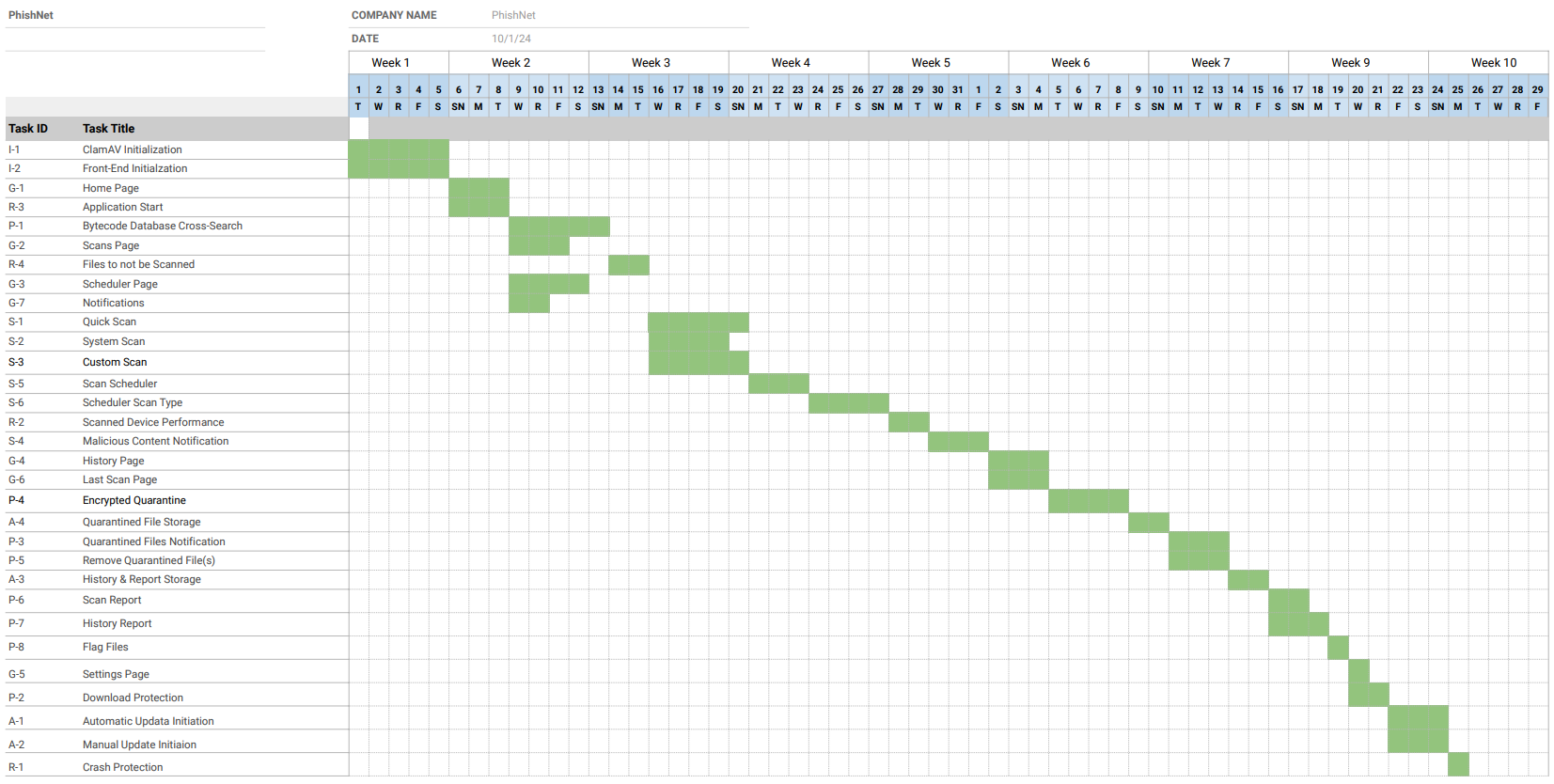
**5.2 Scheduling Graph**



1. **Project Schedule**

**6.1 Schedule**

The diagram outlines the development schedule for PhishNet. “Day 0” marks the starting point corresponding to Tuesday, October 1st, 2024.



The table below represents the schedule, which includes the exact dates and days:

| Task | Start Date | Duration (Days) | End Date |
| --- | --- | --- | --- |
| I-1 | Oct 1, 2024 | 5 Days | Oct 5, 2024 |
| I-2 | Oct 1, 2024 | 5 Days | Oct 5, 2024 |
| G-1 | Oct 6, 2024 | 3 Days | Oct 8, 2024 |
| R-3 | Oct 6, 2024 | 3 Days | Oct 8, 2024 |
| P-1 | Oct 9. 2024 | 5 days | Oct 13, 2024 |
| G-2 | Oct 9, 2024 | 3 Days | Oct 11, 2024 |
| R-4 | Oct 14, 2024 | 2 Days | Oct 15, 2024 |
| G-3 | Oct 9, 2024 | 4 Days | Oct 12, 2024 |
| G-7 | Oct 9, 2024 | 2 Days | Oct 10, 2024 |
| S-1 | Oct 16, 2024 | 5 Days | Oct 20, 2024 |
| S-2 | Oct 16, 2024 | 4 Days | Oct 19, 2024 |
| S-3 | Oct 16, 2024 | 5 Days | Oct 20, 2024 |
| S-5 | Oct 21, 2024 | 3 Days | Oct 23, 2024 |
| S-6 | Oct 24, 2024 | 4 Days | Oct 27, 2024 |
| R-2 | Oct 28, 2024 | 2 Days | Oct 29, 2024 |
| S-4 | Oct 30, 2024 | 3 Days | Nov 1, 2024 |
| G-4 | Nov 2, 2024 | 3 Days | Nov 4, 2024 |
| G-6 | Nov 2, 2024 | 3 Days | Nov 4, 2024 |
| P-4 | Nov 5, 2024 | 4 Days | Nov 8, 2024 |
| A-4 | Nov 9, 2024 | 2 Days | Nov 10, 2024 |
| P-3 | Nov 11, 2024 | 3 Days | Nov 13, 2024 |
| P-5 | Nov 11, 2024 | 3 Days | Nov 13, 2024 |
| A-3 | Nov 14, 2024 | 2 Days | Nov 15, 2024 |
| P-6 | Nov 16, 2024 | 2 Days | Nov 17, 2024 |
| P-7 | Nov 16, 2024 | 3 Days | Nov 18, 2024 |
| P-8 | Nov 19, 2024 | 1 Day | Nov 19, 2024 |
| G-5 | Nov 20, 2024 | 1 Day | Nov 20, 2024 |
| P-2 | Nov 20, 2024 | 2 Days | Nov 21, 2024 |
| A-1 | Nov 22, 2024 | 3 Days | Nov 24, 2024 |
| A-2 | Nov 20, 2024 | 3 Days | Nov 24, 2024 |
| R-1 | Nov 25, 2024 | 1 Day | Nov 25, 2024 |

1. **Monitoring & Reporting Mechanisms**

**7.1 Reporting**

We will deliver our first status report on Thursday, October 27th, 2024. Subsequent reports will be given to Professor Jefferys until we deliver our finished product, which will consist of revised documentation and a presentation with a live demo.

**7.2 Monitoring**

* Phishnet will be monitored by Professor Jefferys.
* Weekly meetings will be held on Saturday nights to discuss our progress following the schedule and our current position.

**7.3 Testing**

PhishNet will primarily be tested using virtual machines (VM(s)). Different Ubuntu 20.04 operating system builds are to be run through multiple VM images. We will perform multiple tests using extremes, such as how much storage will be given to the VM and how the system will behave if Ubuntu is turned off without warning. PhishNet will be tested with different types of files containing malware that will seem legitimate to the human eye. We will test the functions of the directories for quarantining and the directory which will hold all of the records from past scans. Before running our final live demonstration of the software, PhishNet will be tested on a physical device(s) that runs Ubuntu 20.04. Once confirmed that there is a viable demonstration of the software, we will not proceed with further testing of feeding PhishNet files that contain malware as it can not be done 100% safely outside of the VM.

As mentioned in our branching policies, bug, and feature branches produced by each individual member will be confirmed by all team members before being pushed/merged into our GitRepo. The code coverage statistics will be uploaded to our GitRepo once test cases have been developed.