PhishNet

Submitted on 9/19

Version 1.0

Table of Contents

1. Introduction (Aarshia & Julia)

1.1 Purpose (Aarshia)

The purpose of this document is to provide a comprehensive overview of the requirements for the Cybersecurity Anti-Malware Scanning Application (PhishNet). It details the core functionalities, constraints, and system interfaces of PhishNet, addressing key aspects crucial for current and future developers, users, and maintainers. This document serves as a reference for understanding the project's scope, design considerations, and operational guidelines, and it is subject to updates as the project evolves.

1.2 Document Conventions (Julia)

For any future updates to this document, it must adhere to the following:

- Times New Roman font
- 14pt bold font for headings
- 12pt bold font for subheadings
- 12pt non-bolded font for any other text
- Double-spaced font, unless in the case of lists, which use 1.15 spaced font
- Requirements will be tagged in the format [LETTER-NUMBER] and will use the italicized font (i.e. *[A-1]*)

1.3 Scope (Aarshia)

PhishNet is a supplemental malware scanning tool that allows users to scan locations within their Microsoft Windows-based computer for malicious software using the definitions within the ClamAV open-source project. Users can opt to scan the system as a whole, frequently infected directories(hereafter to be referred to as a “Quick Scan”), or a set of user-specified directories. provide malware detection and removal for the Windows

operating system. Additional features include scheduled scans, definition updates, quarantine options, and detailed threat reports.

1.4 Intended Audience (Aarshia)

- Everyday Users: Individuals who use the Microsoft Windows operating system and seek an additional layer of security for their personal computers. These users are looking to enhance their confidence in the overall security of their system by incorporating an extra malware scanner.
- Advanced Users: Individuals who are more experienced with Windows operating systems and require an advanced security solution to address malware on potentially infected systems. This group includes users who might need additional protection or specialized tools to manage and remove malware.
- Technical Enthusiasts and Developers: Advanced computer users interested in exploring the open-source ClamAV project from a different perspective. This audience includes those who wish to understand and potentially contribute to the implementation and development of the ClamAV library through practical application in PhishNet.

1.5 Definitions of Words, Acronyms & Abbreviations (Aarshia)

Words, Acronyms & Abbreviations	Definitions
Git	A distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.

GitHub	A web-based platform for version control and collaboration using Git.
GUI	Graphical User Interface
Open-Source	Refers to software with source code that is freely available for modification and distribution.
PhishNet	A lightweight antivirus/anti-malware tool, available on Windows 10. With a multi-functional scanning tool and an easily accessible history of scan results.

1.6 Software License (Julia)

The license used for PhishNet will be the GNU General Public License, Version 2, in compliance with the licensure requirements of the ClamAV library (LibClamAV). Any 3rd-party frameworks used in PhishNet will be free-licensed and open-sourced as required in the terms of the LibClamAV license.

2. System Overview & Requirements (Gianmarco)

2.1 Product Perspective

PhishNet is an application designed for users seeking to enhance the security of their system. In addition, PhishNet aims to provide a system designed to handle various use cases or scenarios, providing a user experience that reaches as many people

as possible. PhishNet is an open-source project that relies on ClamAV's database for virus signatures, which will help those looking to gain insight into implementing ClamAV. PhishNet is designed for use strictly on Microsoft Windows 10.

2.2 Product Functions

PhishNet is designed to let users select and scan specific disks, directories, or files for scanning, analyzing each file to determine if it contains malware. Moreover, once malicious software is detected, the program quarantines the affected files. Additionally, PhishNet updates virus signature databases automatically, allowing it to identify newer malware threats. The software records all files placed in quarantine, allowing users to review and analyze past malware detection incidents. PhishNet's user interface offers a simplified, one-click scanning along with easy-to-use custom scheduling and selection for scans.

2.3 User Classes & Characteristics

- **Individual Users:** casual or general computer users who want free, reliable, and easy-to-use antivirus software.
- **Technical Users:** advanced users or IT professionals who may need to customize scanning options or review quarantine history for more in-depth threat analysis.
- **Academic Users:** students or independent programmers looking to get involved in an open-source antivirus software project.

2.4 Operating Environment

PhishNet will only run on and be available for users on the Windows 10 Operating System *[O-1]*.

2.5 Design & Implementation Constraints

PhishNet is implemented using C++ *[D-1]* and integrates ClamAV as its score backend for malware detection and scanning *[D-2]*. The user interface will also be developed in C++ *[D-3]* and will provide a native look and feel that seamlessly integrates with Windows 10. The development environment is Visual Studio Code *[D-4]* while testing and debugging will be done using Oracle VM Virtualbox *[D-5]*.

2.6 Assumptions & Dependencies

PhishNet will be developed in C++. Users will have Windows 10 installed and meet the necessary hardware requirements. The software will depend on ClamAV for core virus signatures and bytecode detection. Regular updates and support from ClamAV open-source platform will be essential for PhishNet's detection methods. ClamAV has the following external library dependencies: libcheck, bzip2, zlib, libxml2, libpcre2, openssl, json-c, libmspack, pthreats-win32, libcurl *[D-6]*.

3. External Interface Requirements (Julia)

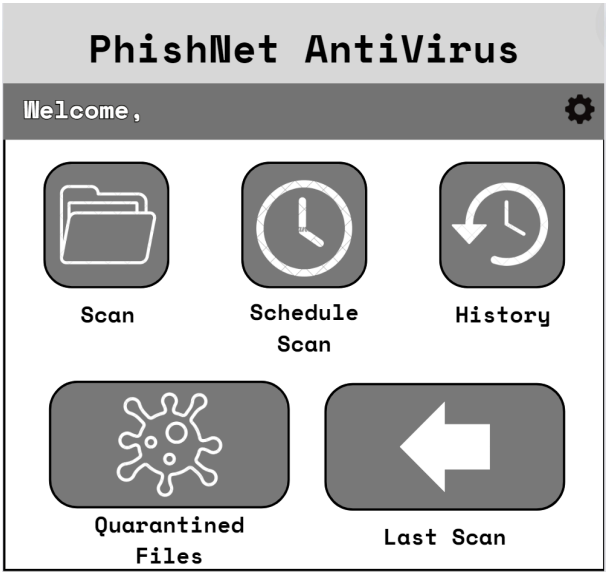
3.1 User Interfaces

Below are images of the current GUI rough mockup for PhishNet. These are not final mockups nor exactly reflect what the final product will look like. This section will be updated as the mockups are edited and finalized.

3.1.1 Homepage Mockup

The user will automatically start on this page every time that the PhishNet application is opened after a restart. The user will have a selection of 5 menu option buttons to click as well as a settings icon in the top right corner to navigate to the 'Settings Page'. The menu selections will be formatted as 3 centered square buttons on top with an icon in the middle of each as well as text below to indicate where the user will be brought. There

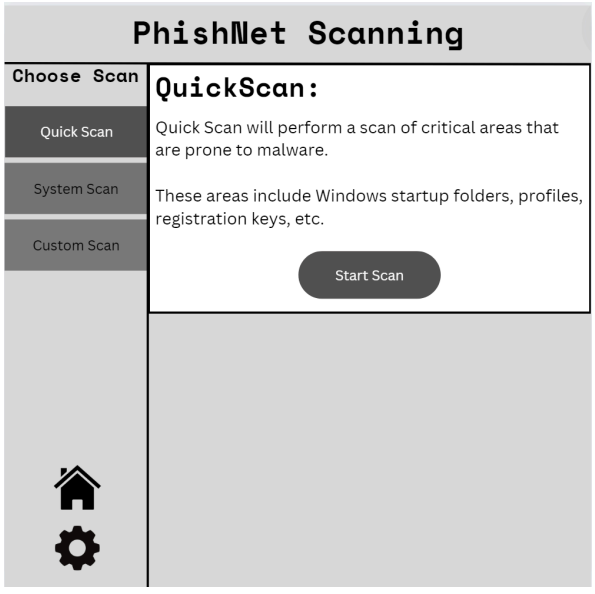
will also be 2 larger rectangular buttons centered below the 3 squares with icons in the middle of each and text below to indicate where the user will be brought.

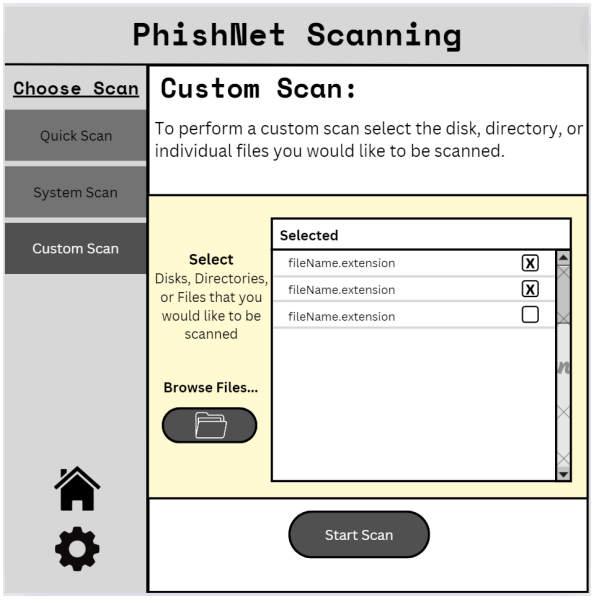


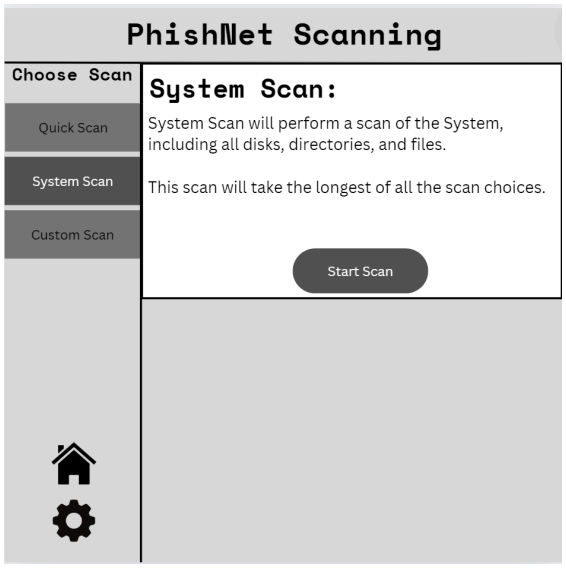
3.1.2 Scanning Options Mockup

The menu to select which scan will be performed will be on the left-hand side of the screen. There will be 3 equal-sized rectangular buttons with text in the middle to indicate the type of scan that the user is selecting. They will be sectioned off by a vertical line which will be about a quarter way into the page from the left. At the bottom of this section will be 2 stacked icons, a home icon on top and a settings icon on the bottom. When selected, the button of choice will change color to a darker shade, and the text will turn white to differentiate the selected button from the not selected. The page will have a default of the ‘Quick Scan’ option selected. This page as well as the ‘System Scan’ page will feature a white box on the top half of the right side of the page. This box will include a header representative of the selection, and body text describing what that option will scan. Below this will be a centered rounded rectangular button that will read “Start Scan”. When clicked this button will initiate the scan of choice.

The ‘Custom Scan’ page will feature the same white box on the right-hand side of the divider, although it will now take up the entirety of the right-hand section. At the top left corner of the white box will read a bolded header reflective of the user choice. Under this text will be body text describing the function of the scan and what actions the user must now take. Spaced and below this text will be another box, pale yellow with a black border. This box will hold text on the upper-middle left side, as well as a rounded rectangular button with a folder icon in the middle, centered below so the user can select what they would like to be scanned. Next to both texts will be another white box with a black border. This box will be split into 2 sections with a vertical black dividing line. The bottom section is much larger than the top. The top section will hold the title of ‘Selected’ and the section below will have a list of what the user has selected to scan. Each selection will be sectioned by a thin black border spanning the width of the white box up until the scroll bar is reached. To the far right of the list will be corresponding checkboxes. Underneath the box will be a centered, rounded rectangular button with the words “Start Scan” in the middle. The user will click this to begin their scan.



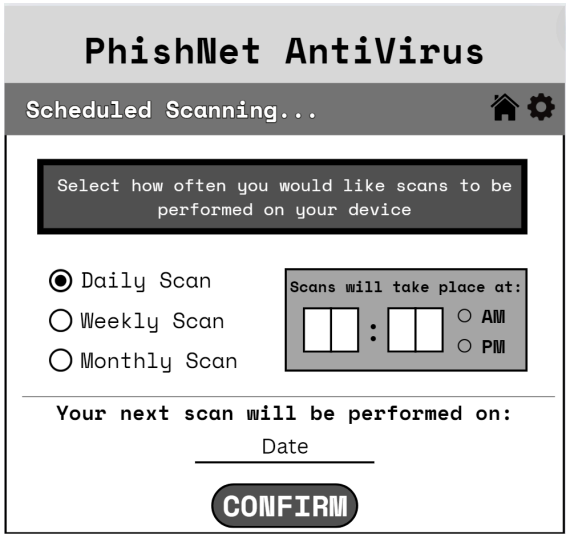




3.1.3 Scheduled Scanning Mockup

This page will feature a header reading “Scheduled Scanning” on top of a wide gray top border which will span the width of the screen. This text will be formatted to the left of the box. On the right side of the gray border will be selection icons/buttons to go to the ‘Home’ page or the ‘Settings’ page. The settings button will be to the right of the home button. Below the border will be a white box that takes up the rest of the page, with thin black borders on the left, bottom, and right sides of the box. Centered at the top of this box will be a dark gray rectangular box with a thick black border with centered white text inside. Below this box on the left side are 3 radio buttons arranged vertically, with text to the right of each to choose which schedule they would like to follow. Below this selection is a very thin black border spanning most of the width of the screen. Under the divider will be a bolded centered subheading. Under this will be a small black line in the center of the screen which will hold the text of the next date a scan will be performed. On the bottom of the screen will be a dark gray, rectangular, rounded button with the text

“CONFIRM” in the middle. The user will click this button to confirm their selections and finalize the schedule for their upcoming scans.

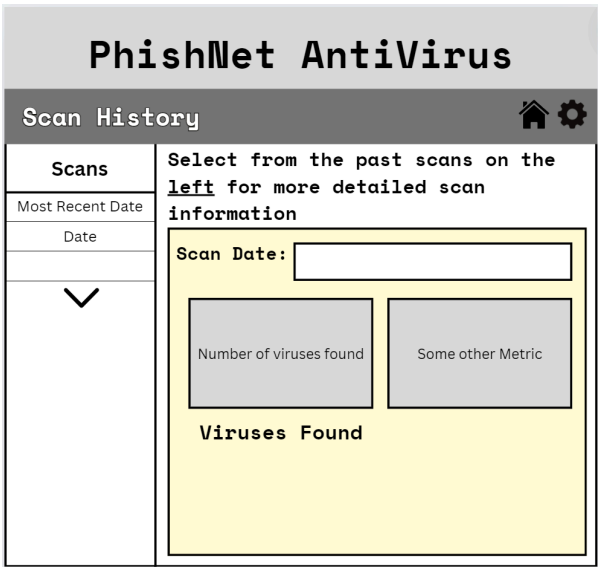


(Need to add which type of scan)

3.1.4 Scan History Mockup

This page will feature a left-formatted header reading “Scan History” on top of a wide gray top border which will span the width of the screen. This text will be formatted to the left of the box. On the right side of the gray border will be selection icons/buttons to go to the ‘Home’ page or the ‘Settings’ page. The settings button will be to the right of the home button. Below the border will be a white box that takes up the rest of the page, with thin black borders on the left, bottom, and right sides of the box. There will be a vertical divider about a quarter into the page from the left. In this section there will be a bolded header reading “Scans” and under this will be a divider then a selectable list of all of the dates of the scans that have been performed from most recent to least. On the right side of the divider, on the top left part of the section, there will be bolded body text giving instructions on how to select a scan. Under the text will be a centered pale yellow box with a thin black border, this box will not take up the width of the right side. The top left of this box will have bolded text reading “Scan Date” and to the right of it, there will be a

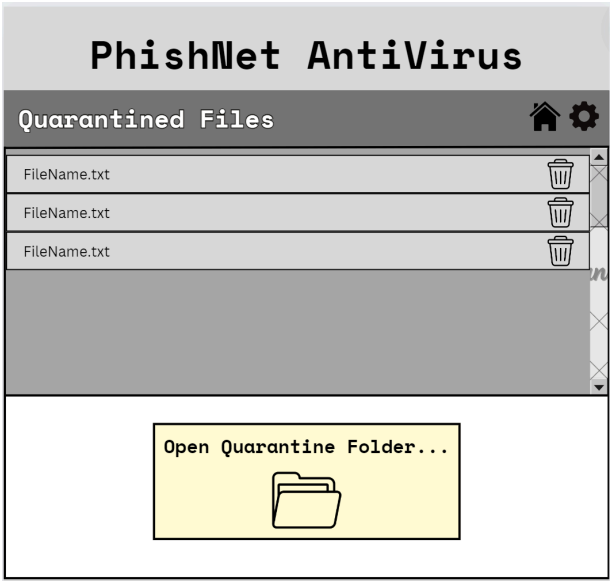
smaller rectangular white box with a black border. Inside this box will have text reflecting the date of the scan chosen from the left. Under this section, there will be 2 light gray rectangular boxes with thin black borders, with a number in the middle and text underneath to show what the numbers represent. The user will be able to click these boxes for more information.



3.1.5 Quarantined Files Mockup

This page will feature a header reading “Quarantined Files” on top of a wide gray top border which will span the width of the screen. This text will be formatted to the left of the box. On the right side of the gray border will be selection icons/buttons to go to the ‘Home’ page or the ‘Settings’ page. The settings button will be to the right of the home button. Below the border will be a white box that takes up the rest of the page, with thin black borders on the left, bottom, and right sides of the box. The top half of the page will have a light gray box with a black border taking up the width of the page. On the far right side of the box, there will be a scroll bar for the section. Within this box, there will be thinner light gray boxes that will contain the list items of all of the files that have been

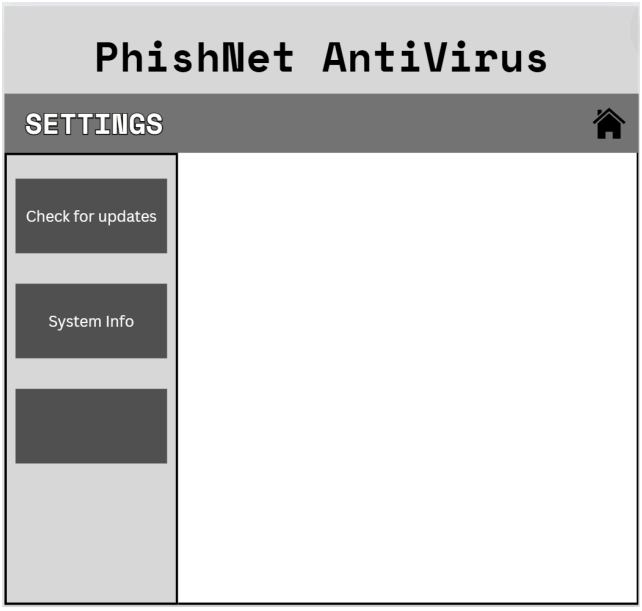
quarantined after a scan. To the far right of these boxes will be trash icons, where the user can delete these files. Underneath the gray box, there is a smaller pale yellow rectangular box that will be centered vertically and horizontally within the bottom right section of the page. The top center of the box will have text reading “Open Quarantine Folder” as well as a folder icon centered below it. The box will function as a button to be able to open the PhishNet directory to see all of the quarantined files in the regular Windows file format.



3.1.6 Settings Mockup

This page will feature a left-formatted header reading “SETTINGS” on top of a gray top border which will span the width of the screen. This text will be formatted to the left of the box. On the right side of the gray border will be a selection icon/button to go to the ‘Home’ page. The settings button will be to the right of the home button. Below the border will be a white box that takes up the rest of the page, with thin black borders on the left, bottom, and right sides of the box. There will be a vertical divider about a quarter into the page from the left. The background of the left side will be light gray and have a thin black border while the right side will have a white background and will only feature

a right and bottom border. The left side will have 3 dark gray rectangular buttons equally spaced from each other. These buttons will not take up the width of the left side and will have text indicating what section will open when they are clicked. These sections will open on the right side of the screen, overlaying the background. This section is still a work in progress.



3.2 Hardware Interfaces (Derived from ClamAV Requirements)

Minimum Hardware Requirements *[O-2]*:

- RAM Requirement: 3 GiB+
- CPU Requirement: 1 CPU at 2.0 Ghz
- Minimum Available Hard Disk Space Requirement: 5 GiB of Free Space in addition to the recommended disk space required for the OS (Windows 10).

3.3 Software Interfaces

PhishNet will require C++ in order to run. PhishNet will use C++ to execute the ClamAV backend and will also utilize C++ in the frontend GUI.

3.4 Communication Protocols (Julia)

PhishNet will require an internet connection for the user to update the application. Updates will almost always include new virus bytecodes. Updates may include GUI changes along with a bytecode update. PhishNet will use https on port 443 to connect to the internet. The user will not be able to check for updates or be notified of a new update release through the application unless they are connected to the internet. If the user is not connected to the internet, the virus bytecodes will remain the same as that of the time of the first installation or the last update made when connected to the internet. A user who does not regularly update PhishNet will be at increased risk for infection as the scan function will not have the newest bytecodes to protect against.

4. Functional Requirements (AJ)

4.1 Scanning Methods

PhishNet will have 3 different options for scans, Quick, System, and Custom. Quick will scan the user-designated main directory that runs the operating system and essential files to the machine *[S-1]*, while System will scan the full file directory of the machine *[S-2]*, and Custom will allow the user to determine what disks, directories, or individual files to scan *[S-3]*. Each scanning method will alert the user of the location of any files found to be malicious *[S-4]*.

4.2 Real-Time Protection

PhishNet will use the ClamAV virus bytecode database to scan files before being downloaded by cross-searching the file with the database *[P-1]*. 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 *[P-2]*.

4.3 Scheduled Scanning

PhishNet will use the windows scheduler to allow the user to choose an interval for scheduled scanning from the options of daily, weekly, and bi-weekly *[S-5]*. The default selection upon software installation will be daily. The scheduled scan will be configured to one of three scan types listed in the scanning methods, with Quick being the default *[S-6]*.

4.4 Quarantine & Removal

When an existing file on the device is deemed to be malicious through scan, PhishNet will quarantine the file and alert the user *[P-3]*. PhishNet will quarantine the file by first encrypting the file and then moving it to a protected folder *[P-4]*, which will prevent further execution of the file. The User will have the ability to remove the quarantined files through the quarantine manager button on the UI *[P-5]*.

4.5 Threat History & Reports

After a scheduled or User-initiated scan, PhishNet will provide a report on any malicious files as well as their location *[P-6]*. PhishNet will also provide a Threat History report, which will allow the user to see which directories in the past have had what malicious files *[P-7]*.

4.6 Updating Virus Bytecode Database

While connected to the internet, PhishNet will connect with ClamAV and prompt the user to perform an update. This will update the virus bytecode database and GUI (if changes are made) *[A-1]*. PhishNet will check for an update before proceeding with any scans to provide the most accurate protection *[A-2]*. A user may also go to the settings page to manually check for updates if the application has not been restarted in a while or a scan has not been performed *[A-3]*.

4.7 Data Management & Storage

Through its own file directory, PhishNet will store Threat History as well as Reports *[A-4]*. PhishNet will also store Quarantined files through a Quarantined file directory in which no files can be executed *[A-5]*.

5. Non-Functional Requirements (Aarshia)

5.1 Performance Requirements

PhishNet is designed to run efficiently on systems meeting the basic Windows 10 requirements: 3 GB or more of RAM, a CPU with at least one core clocked at 2 GHz or higher, and 5 GB of available hard disk space *[O-2]*. As a lightweight antivirus program, PhishNet is not built using large JavaScript front-end frameworks like Electron, ensuring fast performance and a responsive user interface. Users should experience smooth operation without any performance lags or delays when interacting with the software or its GUI *[O-3]*. As the project develops, we will continue optimizing PhishNet's efficiency, potentially reducing system requirements in future versions. Performance benchmarks include minimal latency during user interactions, real-time malware detection, and low resource usage during background scans.

5.2 Security Requirements

PhishNet requires administrative privileges to perform critical operations, including accessing, quarantining, and deleting system files *[O-4]*. These permissions ensure that the application can manage malicious software, even in protected directories. Security measures include:

Administrative Access: Only users with administrative rights will have the authority to review and manage quarantined files, ensuring that unauthorized or accidental deletions are prevented.

Log Access Control: Access to system logs, scan reports, and quarantine history will be restricted to prevent tampering and ensure data integrity.

File Encryption: Quarantined files will be securely encrypted to prevent unauthorized access or execution.

5.3 Software Quality Attributes

PhishNet is designed with a focus on usability, maintainability, and scalability to accommodate both novice and advanced users. Key quality attributes include:

- **Usability**: The software interface is intuitive and user-friendly, with clear prompts, easy-to-understand navigation, and automated workflows. Users will be able to initiate scans, schedule tasks, and manage threats with minimal effort *[Q-1]*.
- **Maintainability**: The codebase and system architecture are modular, allowing for easy updates, bug fixes, and enhancements. Regular updates to the virus bytecode database ensure continued accuracy and relevancy *[Q-2]*.
- **Scalability**: The software is built to scale with system resources, ensuring consistent performance even with large file volumes or intensive scanning operations *[Q-3]*.

5.4 Reliability & Availability

PhishNet must maintain high availability and reliability during operation. The software is expected to function consistently under various system loads and recover gracefully from errors. Key reliability aspects include:

- **Fault Tolerance**: In the event of a system failure or unexpected shutdown, the software will resume the last known state and continue operations without data loss *[R-1]*.
- **Background Operation**: PhishNet will run scheduled and real-time scans in the background without causing performance degradation or requiring system restarts *[R-2]*.
- **System Uptime**: The application is designed to be available 99.9% of the time, ensuring users can rely on consistent protection *[R-3]*.

5.5 Accuracy of Malware Cleaner

PhishNet, utilizing ClamAV's signature detection engine, aims to deliver a high level of accuracy in identifying and neutralizing malware. Regular updates to the virus signature database will enhance detection accuracy over time:

Detection Accuracy: PhishNet will achieve approximately 85% accuracy in identifying known threats, improving as ClamAV’s database evolves *[E-1]*.

Minimization of False Positives: Continuous refinement of detection algorithms will reduce the likelihood of false positives, enhancing user trust and system stability *[E-2]*.

Ongoing Updates: The software will periodically update virus definitions and threat detection parameters to maintain protection against emerging threats *[E-3]*.

6. References

1. ClamAV Documentation. (2024). ClamAV Virus Scanning Engine. Available at: https://www.clamav.net/documents
2. GNU Project. (2024). GNU General Public License, Version 2. Available at: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
3. Microsoft Corporation. (2024). Windows 10 Operating System Specifications. Available at: https://www.microsoft.com/windows10/specifications
4. C++. (2024). C++ Standard Documentation. Available at: https://en.cppreference.com/w/
5. Visual Studio Code Documentation. (2024). Visual Studio Code Development Environment. Available at: https://code.visualstudio.com/docs
6. Open Source Initiative. (2024). Open Source Definition. Available at: https://opensource.org/osd