Vulnerabilities Under Learned Network (V.U.L.N)

Team 4: Hibah, Diego, Kenneth, Justin



Software Requirements Document

Anti-Malware Scanning Application

Version 1.0

**Table of Contents**

**1. Introduction**

1.1 Purpose…………………………………………………………………………………..pg.3

1.2 Scope………………………………………………………………………………...…..pg.3

1.3 Intended Use…...………………………………………………………………………..pg.3

1.4 Stakeholders………………………………………………………………………….pgs.3-4

**2. System Features**

2.1 Malware Scanning Logic………………………………………………………………..pg.4

2.2 File Selection……………………………………………………………………………pg.4

2.3 Malware Detection and Handling…………………………………………………..…pg.4-5

2.4 Quarantine Process………………………………………………………………………pg.5

2.5 Data Management……………………………………………………………………….pg.5

2.6 User Roles……………………………………………………………………………….pg.5

**3. Non Functional Requirements**

3.1 Usability…………………………………………………………………………………pg.6

3.2 Security Requirements…………………………………………………………………..pg.6

3.3 Scalability……………………………………………………………………………..pg.6-7

3.4 Operating Environment Compatibility………………………………………………….pg.7

3.5 Updates………………………………………………………………………………….pg.7

3.6 Deployment and Integration…………………………………………………………….pg.7

3.7 Testing…………………………………………………………………………………...pg.8

**4. User Interface**

4.1 User Interface Design………………………………………………………………..pg.8-11

4.2 User Interface Components………………………………………………………...pg.12-13

4.3 Interactivity…………………………………………………………………………pg.13-14

**5. Interface Requirements**

5.1 Hardware Interfaces……………………………………………………………………pg.14

5.2 Software Interfaces…………………………………………………………………….pg.14

5.3 File Browsing/Communications Interface………………………………………….pg.14-15

**1. Introduction** (Hibah)

**1.1 Purpose**

The specifications for creating the malware detection and removal software are described in the following document. The document will go through several facets of the product, including its key features, conditions, user availability, etc. This document can be revised.

**1.2 Scope**

This software is a local program, with the primary focus on protecting the system of the user against malware by scanning select files on their Linux-based computer, identifying malware, and performing the necessary countermeasures, such as quarantining or depleting infected files.

**1.3 Intended Use**

The software will result in an efficient and simple-to-use malware detection and removal tool that offers essential features to protect user devices. This program will enable users to choose individual files, folders, or disks to scan, update antivirus terms, and retain a history of quarantined items for examination. It will also provide other features such as scheduled scanning.

**1.4 Stakeholders**

* End Users: Individuals using the application to secure their devices.
* Development Team: The group responsible for designing, developing, and testing the software.
* Project Advisor (Professor): Offers direction, criticism, and an assessment of the project's development and results.

**2. System Features** (Hibah)

**2.1 Malware Scanning Logic**

Users are given the ability to set up scans to run every day, every week, or every month. Users can also set an ideal time for their scans. Input from the user initiates the procedure to start. The user-friendly interface of the software allows users to choose the disks, files, or folders that they intend to check for malware on. The scanning engine, in charge of carrying out the malware detection, receives the scan order chosen from the user. The scanning engine examines the documents for malware using various methods, including signature-based detection.

**2.2 File Selection**

Users have the option of selecting certain files, and folders for scanning. For convenient file selection, the application offers a file browser interface. The software creates a scan request with the user-selected files once the user commences a scan.

**2.3 Malware Detection and Handling**

While examining the data, the scanning engine searches for recognized malware characteristics. The response time of the program will be a few seconds and go up to a few minutes when deleting or quarantining a file depending on the size. For every compromised file, it creates a detection report with any harmful code or activity found. The detection report includes details regarding the malware that was discovered, which consists of the impacted file, the type of malware, and the danger level. If numerous malware variants are discovered, separate detection reports will be issued. The software will carry out one of the following actions, depending on the file, if malware is found:

→ Quarantine the affected file(s).

→ Remove the affected file(s).

**2.4 Quarantine Process**

The software's specified lockdown guidelines and procedures are the foundation for choosing to quarantine or remove files. When a file is said to have malware, the system transfers it from the original location to a quarantined location, for the purpose of not harming the system. The user interface of the software notifies individuals of the quarantine action.

**2.5 Data Management**

User personal preferences will be appropriately saved. The past activity of files under quarantine may be viewed by users. The history should show the time and date stamp of the quarantine and will be kept unless specified otherwise.

**2.6 User Roles**

Running manual scans, looking through scan activity history, and accessing account-specific quarantine information are all permissions granted to regular users.

**3. Non Functional Requirements**

**3.1 Usability** (Diego)

V.U.L.N will be an easy-to-perform software that has an interface that makes it easy to know exactly what to do. The software will be able to scan throughout the device efficiently and avoid stopping the flow of the computer while the scan is going on. The scan and all of its settings will be shown next to the button to scan your machine, which shows all the different types of scans and what permissions are allowed. There will be no user training needed in how they will be using V.U.L.N. The software will be so direct and easy to understand that someone will be able to use it to almost near perfection.

**3.2 Security Requirements** (Diego)

V.U.L.N takes its security extremely seriously. When it comes to our security policy with our scanner it is very important to us that we keep security in the user’s hand. The user will first have to permit V.U.L.N to have root access. Then when V.U.L.N gets access it will perform the task and immediately remove its rights back. Whenever V.U.L.N quarantines a file to avoid any further damage that is still up to the user and the user will still have full rights to access the file in a new location. All information on any file and any information on the machine will be saved in the database with the user setting a password to get into the info. The user will be the only person to get into that section of the database's data.

**3.3 Scalability** (Diego)

V.U.L.N will be created with immense scalability and growth in mind. V.U.L.N will use a percentage-based increase within its database storage. When storing information the database will check for the size and increase its size whenever it is close to full to keep itself scaling automatically.

**3.4 Operating Environment Compatibility** (Diego)

V.U.L.N will work on Linux devices and those devices only with plans in the future to implement versions for Windows and Mac. The reason that we chose for our program to run on Linux first is that security and configuring security is a much simpler and overall efficient process compared to Windows and Mac. This way when our program quarantines a file and destroys it, it is much easier faster, and simpler than in other operating systems.

**3.5 Updates** (Kenneth)

V.U.L.N will have an automated update system based on a list of viral patterns. This will be updated by us whenever new patterns are found and then V.U.L.N will have a weekly update that checks for new codes in the list and adds them to the scans.

**3.6 Deployment & Integration** (Kenneth)

* V.U.L.N will utilize the ClamAV Open Source Engine to provide the necessary scanning functionalities. The ClamAV Engine will function as the application’s backend API Gateway and will do the required scanning, storing, and detection of malware info/files via its API server to the database.
* Based on the C++ language and will apply certain libraries, such as the ClamAV library (LibClamAV)

**3.7 Testing** (Kenneth)

V.U.L.N’s capabilities will be tested on computers that run the Linux operating system, and/or Linux virtual machines.

Testing Scenarios:

* Given certain Malware, the application will run and scan (depending on which type of scan the user requests), and detect the respective files. It is then up to the user to decide whether to destroy or quarantine the malware.
* Request for scans to be executed in certain intervals and report how frequently and efficiently the application can perform these actions.

**4. User Interface** (Justin)

**4.1 User Interface Design**















**4.2 User Interface Components**

The components will be made to conduct the following tasks:

* Scan scheduling options
  + This function will be under the settings page which is opened from the table of contents bar on the left-hand side.
  + Users will be able to select the Month, Day, and Time. This can be scheduled for a one-time scan or a series of recurring scans.
* File selection tools
  + There will be a search bar to find a specific file as well as a display that showcases the most recently visited files for convenient access. If the user cannot recall the name of the file a full list view is also available.
  + Once the file is selected, the user simply needs to click scan to start.
* Update the antivirus definitions button
  + The system will check if the version number is the most recent and if it is then a display will show that it is up to date. On the other hand, if the version is outdated the update button will download the patched version.
* Quarantine history viewer
  + The user will have three options for viewing the quarantined files. The first is a list of the most recently added files. The second is a full list of all files quarantined and the third is a search bar in case the user has a specific file in mind.
* Alerts and notifications for malware detection
  + Alerts can be accessed on the home page in the top right corner. Alerts will also pop up to show the results of the scan.
* Help and support resources
  + A support page will be implemented to allow users to submit a formal description report and submit it to the V.U.L.N support team.
* Buttons for selecting different scans
  + Scans will be viewable from the ‘Scanner’ tab in the table of contents. Each scan will have a brief description underneath explaining what method is being implemented.
* Search Bar to select a specific file
  + The search bar will be located in both the ‘File Select’ and ‘Quarantine’ pages to give users a better experience as well as supply them with an efficient way of locating files. To search simply click the search box, type the name of the file, and hit enter.

**4.3 Interactivity/ Scan Engine**

For our front end, we will use HTML and CSS code to construct our user interface. Our team will use MongoDB in the backend to host our databases. API(s) will be created using C++ to communicate with our database(s). The integration of a third-party software scanning engine known formally as ClamAV is crucial to the functionality of our application. The application will be compatible with Linux operating systems with room for future development across other platforms.

**5. Interface Requirements** (Kenneth)

**5.1 Hardware Interfaces** (Kenneth)

Minimum Hardware Specifications:

* 4 GB of disk space
* At most 2 GB of RAM
* 1-2 GHz Processor (CPU)

**5.2 Software Interfaces** (Kenneth)

V.U.L.N requires the Linux Operating system to run currently. A virtual machine would also suffice. C++ will be used to execute the application, as well as the implementation of the ClamAV backend Gateway. A database will be used to store information on any file that the application scans.

**5.3 File Browsing/Communications Interface** (Kenneth)

V.U.L.N will incorporate settings for the user to select what specific files to scan. Certain permissions are requested for this process, these permissions are entirely managed by the user. The UI (User Interface) will provide a simple functionality for anyone to be able to conduct a scan of their choosing. It will also provide a simple overview of the system’s health/status, and give alerts and notifications/malware detections and scheduled scans via the ClamAV backend.

V.U.L.N will keep track of information on files quarantined and destroyed stored on a database. Information collected would be delivered through the help and support resources shown on the Support Page.