James Dempski
Dylan Cenotto
Mark Kornfeld
Thomas Mulvey

Lightweight Antivirus Application (LAVA)

Software Requirements Specification

Table of Contents

1. Introduction

1.1 Purpose (Dylan,Tom)

The purpose of this document is to present an in-depth description for the software application LAVA. As noted in the "Table of Contents", there will be descriptions on the main features, the constraints for ideal operation, and the interfaces of LAVA--along with many other topics of interest for any current developers, future developers, users, or maintainers for this project. This document is to be revised at any time.

1.2 Document Conventions (Tom)

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

- Times New Roman Font
- 14 point bolded font for headings
- 12 point bolded font for subheadings
- 12 point non bolded font for any other text
- Double spaced font, unless in the case of a list which must be 1.15 spaced font.

1.3 Scope (Mark)

LAVA 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. They can opt to scan the full system, frequently infected directories (hereafter to be referred to as a "Quick Scan"), or a set of user-specified directories.

1.4 Intended Audience (Mark)

- Everyday users of the Microsoft Windows operating system who wish to add an additional malware scanner to improve their confidence in the security of their personal computer.
- Advanced users of the Microsoft Windows operating system who may benefit from the additional layer of security, including those looking to remove malware from an infected system.
- Advanced computer users who wish to obtain a different perspective of the open-source ClamAV by exploring a potential implementation.

1.5 Definitions, acronyms, and abbreviations (Everyone)

Word, Acronym, or Abbreviation	**Definition**
Git	Git is 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	GitHub is a global company that provides hosting for software development version control using Git
GUI	Graphical User Interface
Happy Path Tests	In the context of software or information modeling, a happy path is a default scenario featuring no exceptional or error conditions
LAVA	Lightweight AntiVirus Application
Open Source	The term open source refers to something people can modify and share because its design is publicly accessible.

1.6 References (Mark, Tom)

ClamAV:

https://www.clamav.net

ClamAV Library (LibClamAV):

https://www.clamav.net/documents/libclamav

Dear ImGui GitHub Repository:

https://github.com/ocornut/imgui

LAVA GitHub Repository (Currently Private):

https://github.com/kornfeldm/LAVA

Sample System Requirement Specification Document (Template for this Document):

https://gephi.org/users/gephi_srs_document.pdf

Shadowserver Antivirus Day 0 Statistics:

https://wiki.shadowserver.org/wiki/pmwiki.php/Stats/VirusYearlyStats

1.7 Software License (Mark)

LAVA will use the GNU General Public License (GPL) in compliance with the licensing requirements listed for the ClamAV library (LibClamAV). Additionally, as required by the licensing terms of LibClamAV, all 3rd-Party frameworks used in this project will also be free licensed and open-sourced.

2. Overall Description

2.1 Product Perspective (Mark)

LAVA is designed for both those who are looking to better secure their system with an additional layer of security as well as advanced users who wish for more control of the functionality of their security software. In addition, LAVA is designed to offer flexibility both in terms of functionality and in terms accessibility to ensure an experience available to as many users as possible.

LAVA is an open source project, which will afford those hoping to gain a better understanding of ClamAV the opportunity to see a potential implementation. LAVA is designed for use on the Microsoft Windows 10 platform.

2.2 Product Functions (Dylan, James)

The product is designed to detect and deal with different viruses and other malicious software hidden in the users computer while also working in the background. LAVA will constantly update its malware database (if available) to scan with the most up-to-date threat list. It is designed to be user-friendly with a simple Graphical User Interface for our end-users. It can also be used to clean the users computing device to increase performance and increase disk space.

2.3 User Characteristics (Mark, Tom, James)

- General computer users who wish to have a “second opinion” antivirus or a free-open source alternative to other “second opinion” software.

- Students or independent programmers who wish to contribute to an Open-Sourced antivirus software.
- Professionals in the cybersecurity field or ClamAV contributors who would like to see a unique windows GUI for their backend API.
-Users that do not wish to have a large application that drains CPU power and provides features that current virus scanners block behind subscription barriers

2.4 Operating Environment (Tom, James)

As of now, LAVA will only operate on Windows 10. Further OS support (Windows 7, Windows 8, Linux, and macOS) may come at a further date but is not guaranteed.

2.5 Design and Implementation Constraints (Tom)

LAVA is designed in C++. It uses ClamAV as its core backend for scanning. The frontend GUI is built on top of the Open Source framework imgui. The github repo for imgui can be found in the references section. The IDE used for development is Visual Studio.

2.6 Assumptions and Dependencies (Tom, James)

LAVA will be based on the C++ language. LAVA will heavily interface with ClamAV, an open source antivirus engine, which has the following 3rd party libraries as dependencies: bzip2, libcurl, json-c, libxml2, openssl, pcre2, pthread-win32, zlib.

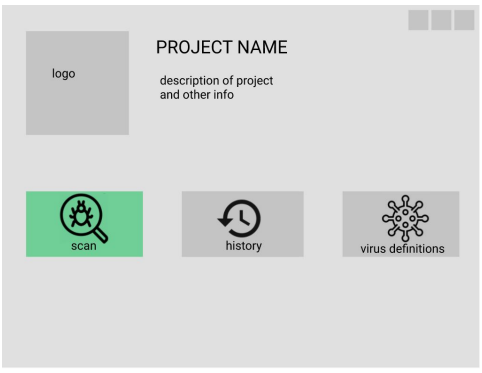
For developers, you will need Visual 2017 (or 2019 with 2017 c++ build tools) and the dependencies shown above.

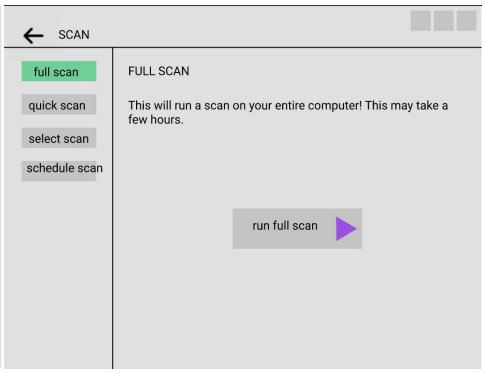
3. External Interface Requirements

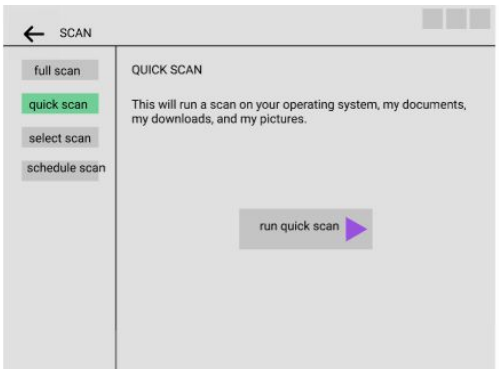
3.1 User Interfaces (Tom)

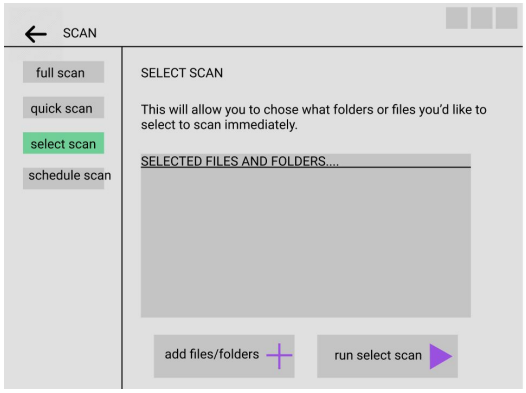
Below are rough wire-frames for LAVA. These are not high fidelity mockups. If the UI/UX changes, the mockups will be updated as well.

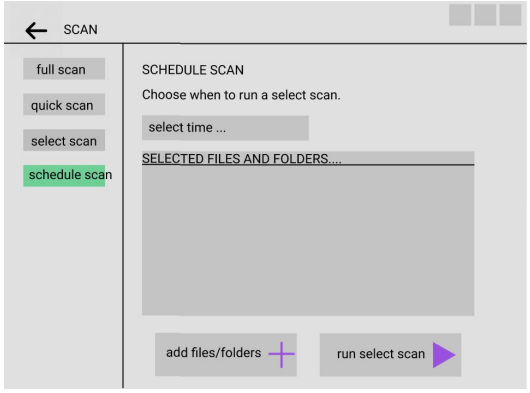
3.1.1 Different type of scans mockup:





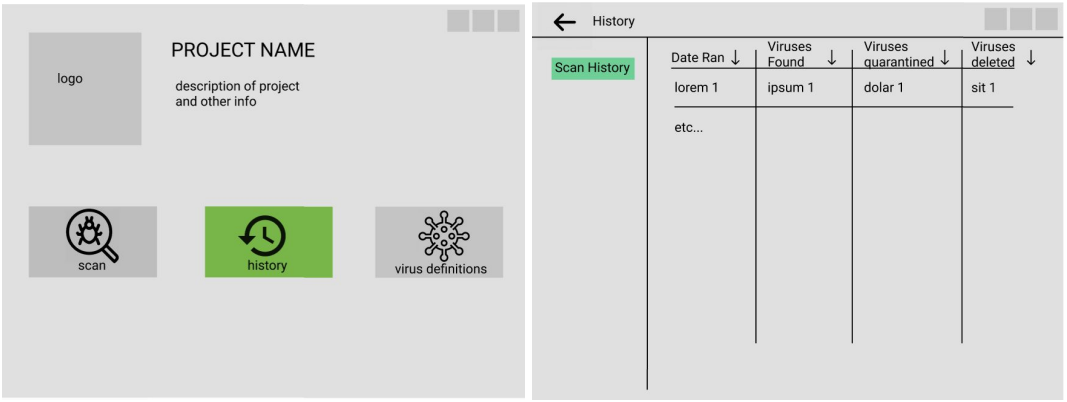






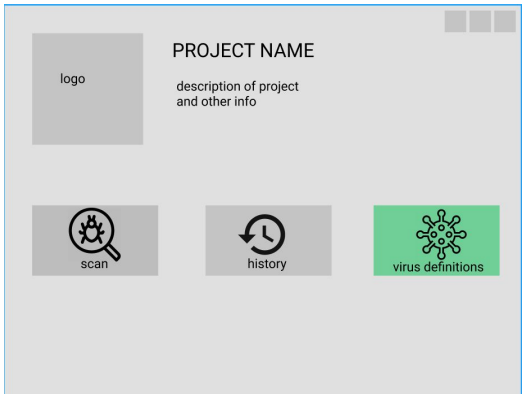
The user will select the left-most icon, scan, and be brought to the next screen. This screen will have a left-vertically oriented menu to select from the different scanning options. The options (features) will be talked more about in section 4, but the main scanning options selectable are: “Full Scan”, “Quick Scan”, “Custom Scan” (which is labeled as “select scan” in the wireframe, and “Scheduled Scan”. The back arrow at the top left will bring you to the main application page with the three “Scan”, “History”, and “Virus Definitions” buttons.

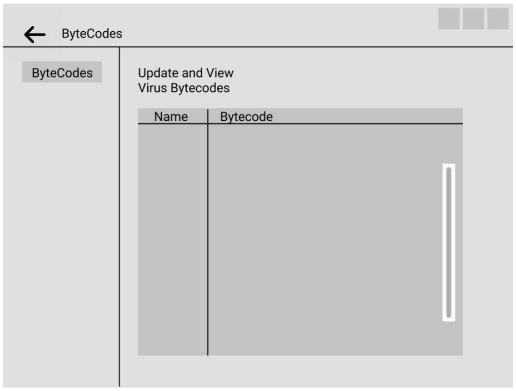
3.1.2 History Mockup



The next button on the home screen is the “History” button. This button when pressed will bring you to a page displaying the previous scan history in a table. It will show when the scan took place, the number of malicious files found, the number of malicious files quarantined, and the number of malicious files deleted. The back arrow button has the same functionality as the scan page, and stays constant throughout.

3.1.3 Virus Bytecode Mockup





The final section on the home page is the “Virus Definitions” home page. This is where the user will be able to view the internal database of “virus bytecodes”. Users will be able to add new definitions here manually if they please. This database of bytecodes will be updated before every scan.

3.2 Hardware Interfaces (Derived from ClamAV Requirements)

Minimal requirements:

3 GiB or more of RAM

1 CPU of 2GHz or more

5 GB or more of free hard disk space

3.3 Software Interfaces(Tom, James)

LAVA will require C++ to run. More specifically, C++ will be used in the design of both the GUI and the execution of the ClamAV backend. The code will be compiled with Visual Studios 2017 Build Tools just like ClamAV.

3.4 Communications Interfaces (Tom)

LAVA will require an internet connection to update virus bytecodes. It will use https on port number 443. If no internet connection is available, it will not be able to check if any new updates are available, thus leaving the end-users at a higher risk by not having the most up to date bytecodes. The only bytecodes that will be available will be locally from either the first installation or last update (if applicable).

4. System Features

This section will outline LAVA's system features (functional requirements). The order is arbitrary and not of any importance or ranking. If new features are added, thought of, or changed, those updates will be displayed here.

4.1 Quick, Complete, and Advanced Scanning (Dylan, James, Mark)

- 4.1.1 Different Levels of Scanning
    - LAVA will have 3 different levels of scans. Quick, Full, and Advanced (also referred to as "custom").
- 4.1.2 Quick Scan Capability
    - The Quick Scan will search through the main directories that configure the operating system and that are essential to the machine's function
    - The Quick Scan will also search through the "malware-prone" directories. These directories begin as a predefined set of folders commonly infected. As malware is detected in various locations, the "malware-prone" directories scanned during a Quick Scan will be updated to provide extra emphasis on directories deemed "vulnerable." These "malware-prone" directories will be kept track of by the Antibody File, which is described in further detail in *Section 4.3*.
- 4.1.3 Complete Scan Capability
    - The Complete Scan will recurse through every file and directory on the machine, including attached storage.
- 4.1.4 Advanced Scan Capability

    - The Advanced Scan will allow more experienced users to scan by specific file or directory, including a set of directories. A system explorer will allow users to easily select which file or directory/directories they wish to scan. It will also have a browser cache cleaning option.
- 4.1.5 Maintain System Integrity
    - LAVA puts emphasis on security without potential detriment to the reliability of the user's machine. As such, LAVA will not delete or quarantine files that are essential to the system. The program will only let the user know that there is malicious software in the system files. This is true for any scan done using LAVA. For example, if a file located in C:/WINDOWS is detected to be malicious, LAVA will only alert the user of the potential problem as deleting certain files required by the Windows operating system may cause stability issues or even crashes.

4.2 Scheduling (James)

- 4.2.1 Scheduled Scans
    - LAVA will use Windows Scheduler to run scheduled scans and this option can be configured to occur daily, weekly, or monthly.
    - This option can also be configured to run any of the 3 scans that LAVA provides as well as target specific directories.

4.3 Antibody File (James)

- 4.3.1 Maintaining of an Antibody File
    - After the initial scan and quarantine, the directories that contained the malware will be stored in the Antibody File
    - As more scans happen, this Antibody File will populate and these directories will be marked for future, optimized scans such as the Quick Scan.

4.4 Malware Database Updates (Dylan)

- 4.4.1 Updates of Virus Bytecodes
    - LAVA will update its antivirus definition database every time a scan is launched.
    - If there is no valid internet connection, this step will be skipped.

4.5 Working in the Background (Dylan)

- 4.5.1 Boot Startup Opt-Out
    - LAVA will turn on at system boot up and consistently run in the background unless the user disables that function.
- 4.5.2 Real-Time protection

    - LAVA will check newly downloaded files for existing virus bytecodes and quarantine them, if necessary. It will alert the user as well.
- 4.5.3 External Drive Protection
    - LAVA will check connected USB drives for any malicious software and quarantine the infected files or eject the drive completely.

4.6 Support (Dylan)

- 4.6.1 Continued Systemed Support
    - The application will also have a support section, where the users can report bugs or issues that they have come across while using LAVA. These reports will be automatically sent to a support email, where it can be reviewed by the support team.

5. Other Nonfunctional Requirements

5.1 Performance Requirements (Tom)

LAVA will need the basic Windows 10 requirements to run. That is, 3 GiB or more of RAM, at least 1 CPU core clocked at 2GHz or more, and 5 GB or more of free hard disk space. LAVA is meant to be lightweight, and is not built on any large Javascript front-end frameworks (like Electron) so the antivirus program will be quick and responsive. There should be no performance “hiccups” or latency when operating the software and interfacing with the GUI. As the project evolves, we will attempt to maximize the efficiency of LAVA, which may potentially allow us to drop software requirements in the future.

5.2 Security Requirements (Tom)

LAVA will need “System/Administrator” privileges in the local Windows User Groups. The program will need to isolate (quarantine) files and delete others and check extended

information across the filesystem. The users will be prompted to run in Administrator mode before opening the program like any other Windows Application.

5.3 Software Quality Attributes (Tom)

The UI of LAVA is expected to be simple and intuitive for any user. Each feature will be designed to fit this criteria. Every feature will be manually QA-ed from members of the team, and there will be “Happy Path Tests” documented for QA for each main feature. Code coverage tests will be provided on the github repo for the “master” branch. Each code update to the “master branch” should be done with the intent of increasing user experience.

5.4 Accuracy of Malware Cleaner (Mark)

Although there is no specified requirement for antivirus accuracy, LAVA aims to achieve about 75% accuracy. The windows version of ClamAV, from which this product is based, is typically ranked in the mid to low end of security software rankings such as Shadowserver’s Day 0 test. However, because this product is created for the purpose of using it as a “second opinion” scanner, we believe the accuracy necessary to produce a more-than-competent product will be offered. Additionally, ClamAV is an open-source project that will improve over time. Due to LAVA’s use of ClamAV’s API, LAVA will improve on its own along with ClamAV.