Software Requirements Specification

for

Virus Removal System (VRS)

Version 1.0 approved

Prepared by Yash Saxena, Vatsal Patel, Moksh Talreja, Tulika Singh

NIIT University

25 September 2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Virus Removal System (VRS) is an anti-virus software that does not rely on virus definitions, i.e. the virus dictionaries that get updated on the regular anti-virus databases. In addition to the regular virus dictionary check, VRS also detects suspicious software and virus using real time monitoring of the system properties during the software's run.

1.2 Document Conventions

No conventions are used in this document.

1.3 Intended Audience and Reading Suggestions

The audience intended for this software's use are all people who use Windows systems on a regular basis and would like to upgrade to an antivirus software better capable of handling the security of their systems. VRS can also be modified to work on servers and corporate systems.

A survey was conducted online amongst the age demographic of 18-30 years regarding their personal experiences with Antivirus Softwares. The results of the survey can be viewed here.

1.4 Product Scope

In scope:
-VRS can find out whether there is a sudden fluctuation in performance parameters of a service or a software and depending on the degree of fluctuation, it can term the respective service suspicious and give the user an option to abort it.
-VRS is also hierarchically built on a regular antivirus, because if it detects a file name which matches the virus dictionary, it will immediately clean the virus off the user's system.

Out of scope:
-VRS cannot for sure pinpoint a virus out, it can only say so probabilistically.

1.5 References

No links are referred in this document.

2. Overall Description

2.1 Product Perspective

VRS is a standalone software, built freshly as an initiative for the group's Software Engineering Project.

2.2 Product Functions

-The product classifies software as suspicious or not suspicious using real time monitoring of the software's performance and comparing it with previously collected data.
-The product also immediately quarantines files which match any entry in the virus dictionary.
-The product gives the user an option to abort software which it terms out as suspicious.

- The product should support the following use cases:

Use Case	**Description of Use Case**
Installation	Creates and initializes working files
Start scan	Analyse data to report behaviour of suspicious files
File Operation	Decide whether to keep a file and mark it as trusted or delete a suspicious file.
Run Simulation	Provide a virtual environment and examine a process/service file.

2.3 User Characteristics and Principal Actors

-This software can be used by all equally, regular PC users and corporations alike.

-The principal actors in VRS are "user" and "system".

2.4 Operating Environment

Operating System: Windows 7 or above
Processor: i3 or above
Ram: 4GB or above
Hard Disk: Minimum 2GB (May require more space as time goes on)

2.5 Design and Implementation Constraints

Only new operating systems will be able to run this software efficiently, as real time monitoring of all services takes place and the data gets stored.

2.6 User Documentation

All documentation and manuals will be available inside the software only.

2.7 Assumptions and Dependencies

No assumption has been taken during the construction of this product. This product depends on various programming tools such as gcc, make tool etc.

3. External Interface Requirements

3.1 User Interfaces

The primary interface is ASCII text which is seen on the command line. A Graphical User Interface is currently under development.

3.2 Hardware Interfaces

Hardware interaction is basic and can be accomplished through keyboard and mouse.

3.3 Software Interfaces

Software interfaces are seldom used. Only a connection with the operating system yields the performance parameters of each software during its run, which is accurately accomplished through invoking functions provided with the C library.

3.4 Communications Interfaces

No communication interfaces are used in this product.

4. System Features

Various features are available in VRS.

4.1 Detection of malicious files.

4.1.1 Description and Priority

Any filename which matches a record in the dictionary is immediately removed.

4.1.2 Stimulus/Response Sequences

This feature will be invoked during the inception of any new file, specially downloaded files. The user also has an option of running security whenever he feels to do so.

4.1.3 Functional Requirements

This feature works by spidering through the file system and matching each filename with those of record entries.

REQ-1: Access to all filenames and folders.
REQ-2: Authorization to remove malicious files.

4.2 Detection of possibly malicious files.

4.2.1 Description and Priority

Any file/service/software which fluctuates from its regular performance parameters is termed suspicious. The user, in such a case that a file is termed suspicious, is immediately prompted with an option to abort the file/service/software.

4.2.2 Stimulus/Response Sequences

This feature will continuously run during VRS' run. It will keep track of all the softwares running and keep updating its performance parameters in real time.

4.2.3 Functional Requirements

This feature works by obtaining performance parameters of any software running, and recording its performance in its database. Any fluctuation from its regular performance leads to terming that file suspicious.

REQ 1: Access to performance parameters
REQ 2: Prompting mechanism to inform user

5. Other Nonfunctional Requirements

5.1 Performance Requirements

In order for this product to run successfully and efficiently, it is recommended that it is installed from the very beginning on the operating system. This way the results will be highly accurate.

5.2 Safety Requirements

A possibly important file for the user, can be removed if it matches a dictionary entry.

5.3 Security Requirements

All data created by the product will be protected from all software access, as any modification to that data by a third party software can lead to erroneous running of the product.

5.4 Software Quality Attributes

This software is especially useful for those systems which are not only infested by virus but also are having performance issues, for eg. (low memory, slow performance)

5.5 Business Rules

This software is closed source. Any attempt to modify it, or change it anyhow can lead to legal conflict.

6. Other Requirements

No other requirements for this product.

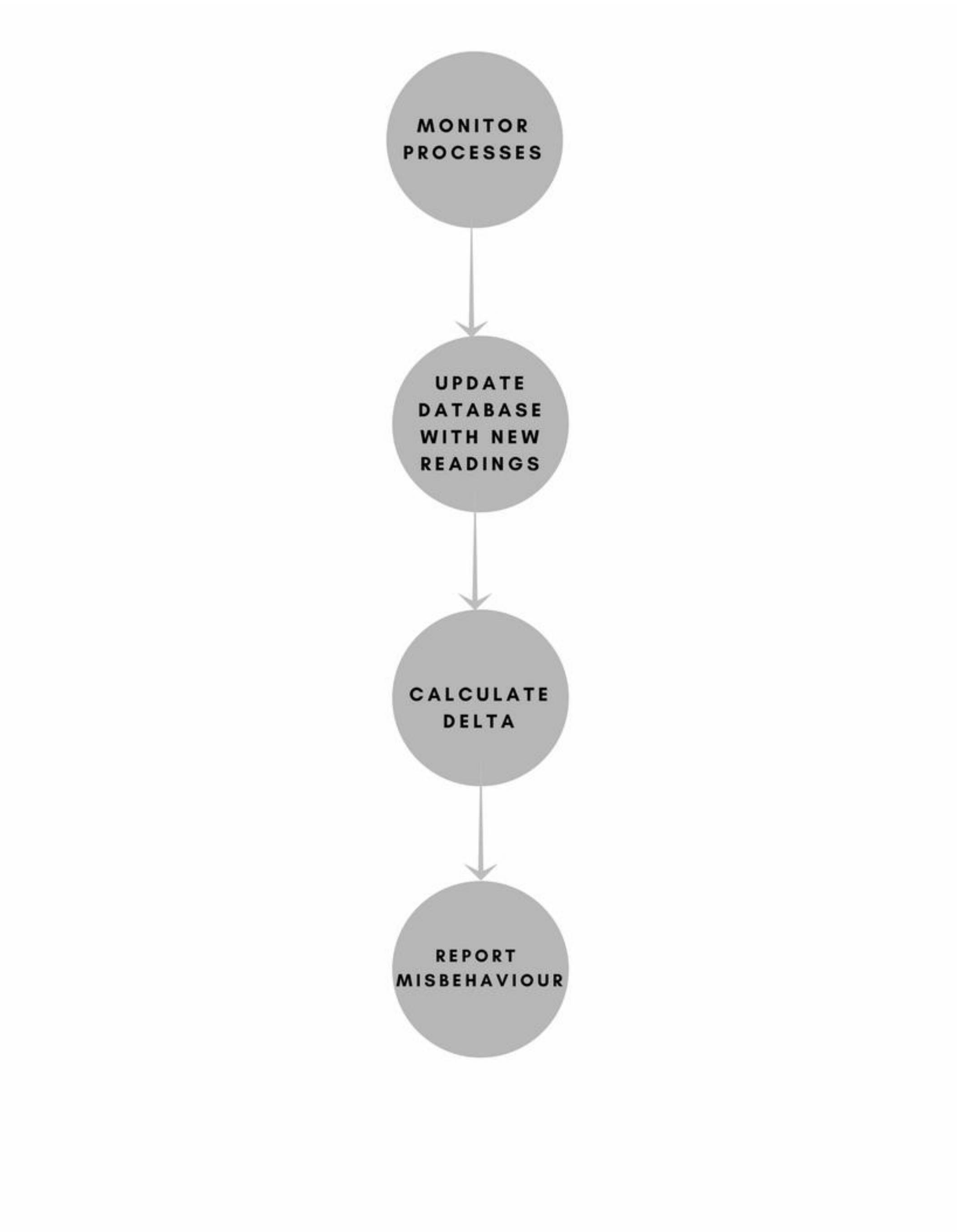
Appendix A: Glossary

VRS: Virus Removal System
Suspicious software: Software which can be termed as possibly containing malware
Virus Dictionary: A list of names of files which are confirmed malicious

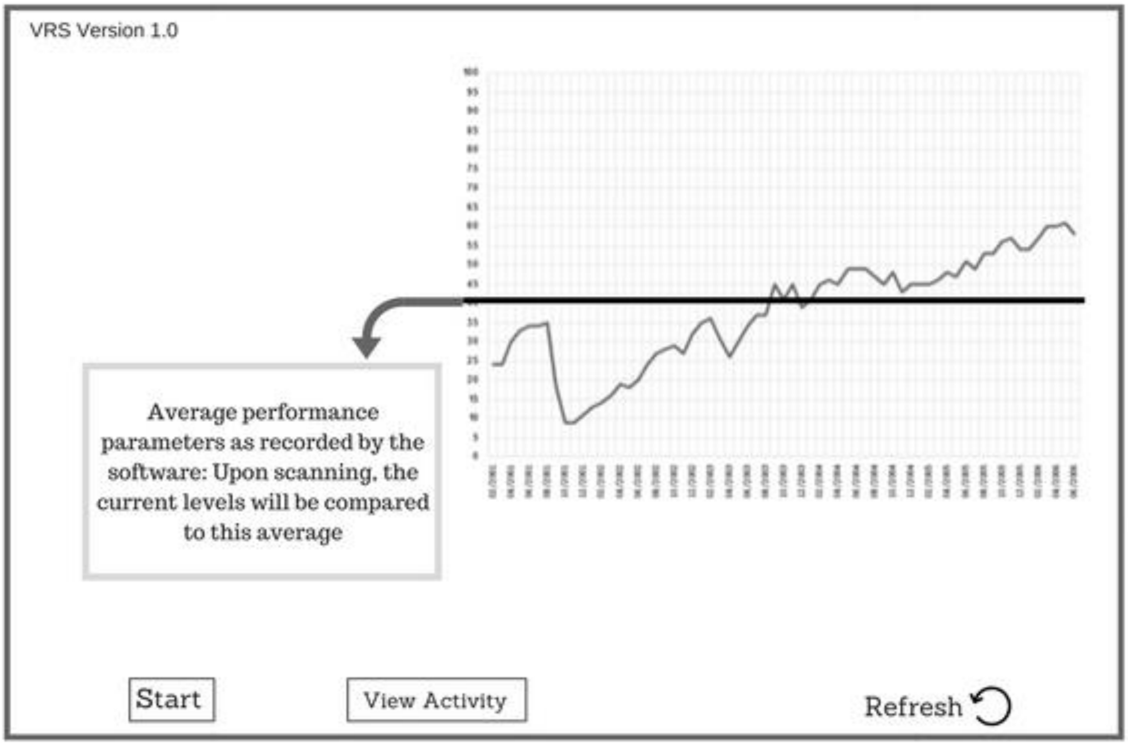
Appendix B: To Be Determined List

A virus dictionary is under construction at the time of writing.

Appendix C: Analysis Model



Appendix D: User Screen



Average performance parameters as recorded by the software: Upon scanning, the current levels will be compared to this average

- This is a basic blueprint of what the application will look like when run. UI Modifications are under progress.
- The "View Activity" button will open a screen which shows detailed graphs of each of the processes individually.