Simple Malware Analysis Reporting Tool (S.M.A.R.T.)

By

Alex Winkler, Yashovar Chilupuri, Omar Hill, and Prateek Chellani

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology/Cybersecurity

Alex Winkler	11/29/21
Alex Winkler	Date
Yashovar Chilupuri	11/29/21
Yashovar Chilupuri	Date
Prateek Chellani	11/29/21
Prateek Chellani	Date
Omar Hill	11/29/21
Omar Hill	Date
Tyler Hopperton	11/29/21
Tyler Hopperton	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2022

Table of Contents

List of Illustrations

Tables

Figures

Abstract

Given the ever-changing nature of the technology field, the demand for cybersecurity and malware analysis is skyrocketing. However, in a field where time is of the essence, several hour-long malware analyses cannot be the norm. Despite this, most dynamic malware analyses take an hour on average, and faster solutions are out of the budget of most small to medium sized companies. SMART combines several popular open-source tools into a simple and intuitive UI to allow users to conduct static and dynamic analyses within minutes. The SMART front-end allows users to upload files or URLs, which are then passed through APIs to our tools. These tools produce individual reports, which are then scraped, and the user sees a single, integrated report with all the requested information.

Introduction

Project Summary:

Simple Malware Analysis Reporting Tool (S.M.A.R.T.) integrates all the essential tools needed to analyze a file so that users can analyze files in a secure and risk-free manner. The platform will be hosted on a website and will allow users to upload any files, IPs, and URLs. This file will be scanned for potential security issues or vulnerabilities, against a variety of known security databases. In addition to this, the web app will integrate a lot of the industry-recognized tools for file analysis, such as VirusTotal, PeStudio, Cuckoo Sandbox, and OpenVAS to allow Users to carry out further detailed analysis.

Problem Statement:

With the recent advancement of technology, there is an increased necessity for companies and organizations in every field to interact with the internet. While this has proven to be an excellent way for the advancement of technology, it is also the primary source of a new type of threat – Cyber Attacks. A 2015 report from the Ponemon Institute reported that of 350 multi-national companies surveyed, all of them reported having been the victim of a malware attack, and on average were breached 1.3 times a week. Several other studies have proven that the source of most malware attacks roots from a corrupted file downloaded by an employee or stored on the corporate system.

Online malware analysis allows for safer and more efficient ways of analyzing malware due to its decentralized nature. Dynamically analyzing malware on computers requires executing the malware which could cause damage or expose valuable information. Thus, using online tools

operating on virtual machines enables users to analyze malware without any risk. Existing market alternatives address this problem either by using static analysis or a combination of static and dynamic analysis which requires paid subscription or a limited number of attempts. In the paper “An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis" by Datta et al, the authors suggest that the existing tools take variable amounts of time to complete a malware detection, however, they also explore the reasons why this may be the case. On average about an hour is considered quick but this depends on the detection method used. In Cybersecurity now, time is of the essence, and therefore faster, reliable, and relevant analysis is essential.

Solution:

Simple Malware Analysis Reporting Tool (SMART) seamlessly integrates popular open-source malware analysis tools such as Cuckoo Sandbox, PeStudio, and VirusTotal into a single, intuitive UI hosted over a single web application. Utilizing a combination of static and dynamic analysis, SMART is capable of reliably analyzing user uploaded files, URLs, or IPs within minutes and presenting its findings in a descriptive report. With the use of SMART, cybersecurity professionals can save a significant amount of time on malware analysis, solving perhaps one of the most challenging problems in the industry today.

Project Source:

The project idea came up in a brainstorming session after several previous ideas that tried to address problems that we were not necessarily primary users of. However, having worked in security jobs, we know the importance of file analysis, and currently, users must hop through

several websites to analyze a single file. Having experienced this tedious task ourselves, we wanted to find a reliable way to integrate everything into one, secure box.

Discussion

Project Objectives/Goals:

- The web app will conduct static and dynamic analysis of any file that is uploaded and generates a report detailing the findings.
- The web app will utilize VirusTotal and PeStudio for static analysis, and Cuckoo sandbox for dynamic analysis.
- **Reach Goal:** If time permits, the web app will also utilize OpenVas and exploitDb to further enhance the analytical relevance.
- **Reach Goal:** An API will be included to aid further development and customization.

Project Scope:

In this section include a comprehensive scope of your solution along with a quick outline of the solution. The scope should be written out into actionable items that can be further broken down when developing a project plan.

“Our team will develop a functional application that enables users to solve xyz problem by utilizing the following features and functionality.”

Quick Project Timeline:

Table 1: Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1	Team Contract	1 week	09/13/2021	09/20/2021
2	Complete project plan		09/20/2021	10/01/2021
3	Website Front End	2.5 weeks	10/01/2021	10/18/2021
4	Website Back End	2.5 weeks	10/01/2021	10/18/2021
5	VM hosting	1.5 week		
6	Learning tools	1 week	09/24/2021	10/1/2021
7	Integrate VirusTotal	2 months	01/10/2021	12/12/2021
8	Integrate PeStudio			
9	Integrate Cuckoo Sandbox			
10	Combine tools			
11	Alpha Testing	2 weeks	01/10/2022	02/24/2022

12	Alpha Revisions	2 weeks	01/24/2022	02/07/2022
13	Beta Testing	1 week	02/07/2022	02/14/2022
14	Final Revisions	1 week + buffer	02/14/2022	02/25/2022

Table 1: Quick Project Timeline

Technologies Used:

PeStudio

VirusTotal

OpenVAS

Cuckoo Sandbox

Website hosting - AWS

GitHub – shared repo

Python – scripting

Typescript

CSS

Technical Architecture Diagram:

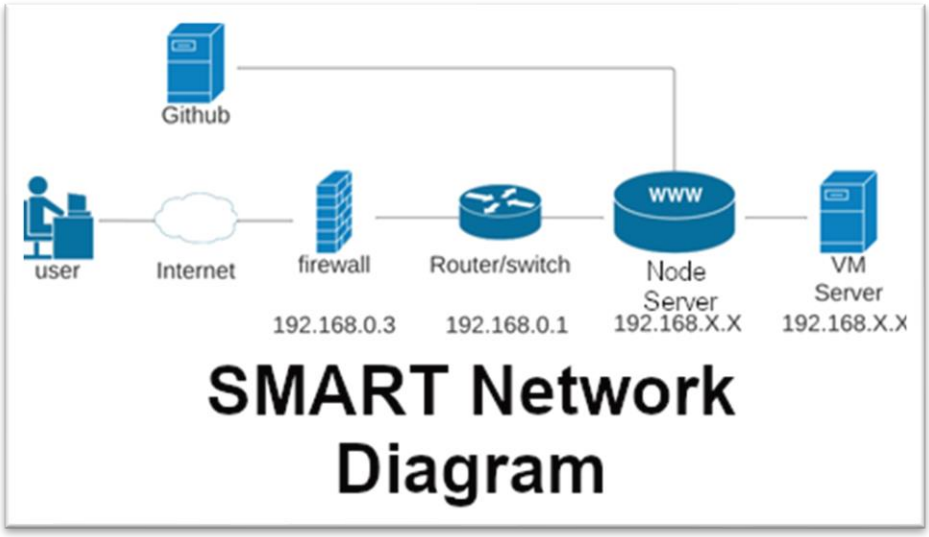


Figure 1:SMART Network Diagram

User Personas:

The following are the customers that the product is focusing on to satisfy their needs in regard to cyber security and malware analysis.

Table 2:User Persona

<table>
<tr><td colspan="2">User Persona: 1</td></tr>
<tr><td rowspan="4">  </td><td>Title

Cyber Security Analyst</td></tr>
<tr><td>Name

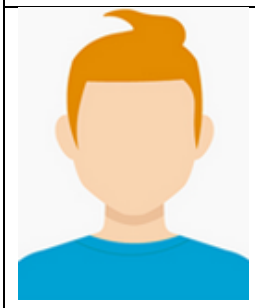
Peter</td></tr>
<tr><td>Age

28</td></tr>
<tr><td>Gender

Male</td></tr>
<tr><td>Behavior</td><td>Peter works at a financial institution and is required to monitor and analyze any software or file that is brought onto to the system to approve it. He receives requests for software approvals every week and sometimes needs to analyze files that are sent as attachments in phishing attempts to the employees of the company.</td></tr>
<tr><td>Pain</td><td>Tight deadlines regarding malware analysis at work, and lack of tools to quickly and insightfully analyze files, programs and software.</td></tr>
<tr><td>Needs & Goals</td><td>• Wants to ensure cyber-safety of company.</td></tr>
</table>

	• Wants to be able to analyze potentially malicious files quickly and reliably. • Requires a tool to aid him to do so.

Table 3:User Persona 2

User Persona: 2	
	Title Cybersecurity Enthusiast
	Name Josh
	Age 21
	Gender Male
Behavior	Josh is a student majoring in computer science and has a keen interest in cybersecurity. He likes to make sure anything he downloads from the internet is safe. He relies on antivirus but wants a tool that can provide a report that is readable and provides technical information regarding the threat actors so he can learn.
Plan	Wants to be able to analyze files and software in a secure and cost-effective manner.

	Not extremely familiar with technical jargon and most existing tools provide analysis he cannot use.
Needs & Goals	• Wants to ensure security of files before installing on his devices. • Needs a tool that is accessible and has an easy interface to use. • Aims to learn more about cybersecurity by analyzing potentially malicious files on his computer.

Use Cases:

The following are the potential use cases which portray scenarios in which the users can reach their desired solutions using S.M.A.R.T.

Table 4:Use Case ID 1.1

Use Case ID	1.1
Use Case Name	Analyze a file
End Objective	To see if the file uploaded is malicious
User/Actor	Cyber security analyst
Trigger	New software introduction to the environment
Frequency of Use	Almost daily
Preconditions	A request for new software has come in
Basic Flow	1. A request for new software has come in

	2. The cyber security analyst has to analyze and approve the software to be introduced into the network. 3. The cyber security analyst extracts the .exe file and uploads it on to SMART 4. SMART produces a report regarding any malicious indicators. 5. The cyber security analyst reads the report and decides if there are any indicators of concern. 6. The report yielded no malicious indicators. 7. The cyber security analyst decides the software is safe for installation. 8. The cyber security analyst approves the software for installation.
Alternate Flow	1. The report yielded three known malicious indicators. 2. The cyber security analyst decides the software is not safe for installation. 3. The cyber security analyst does not approve the software for installation.
Postconditions	There is no malicious software that is introduced to the network.

Table 5:Use Case ID 2.1

Use Case ID	2.1
Use Case Name	Analyze network connections
End Objective	To ensure security of network and open ports.
User/Actor	Cyber security analyst
Trigger	New network connection request
Frequency of Use	Weekly
Preconditions	A request for a new network connection.
Basic Flow	1. A request for a network connection has come in 2. The cyber security analyst must analyze and approve the network before accepting it. 3. The cyber security analyst identifies the network and uses SMART. 4. SMART produces a report regarding any malicious indicators. 5. The cyber security analyst reads the report and decides if there are any indicators of concern. 6. The report yielded no malicious indicators. 7. The cyber security analyst decides the report is good and the network is safe

	8. The cyber security analyst does approves the network connection request.
Alternate Flow	1. The report yielded known malicious indicator for the network. 2. The cyber security analyst decided against connecting.
Postconditions	There is no malicious software that is introduced to the network

Table 6:Use Case ID 3.1

Use Case ID	3.1
Use Case Name	Analyze malicious indicators
End Objective	To see if the file uploaded is malicious
User/Actor	Cyber security enthusiast
Trigger	File malware analysis
Frequency of Use	As needed, expected to be multiple times a week.
Preconditions	Downloading a new file onto the computer
Basic Flow	1. A new file is to be downloaded onto computer 2. The cyber security enthusiast wants to analyze the file and see if there are any malicious indicators. 3. The cyber security enthusiast uploads a file to SMART 4. SMART produces a report regarding any malicious indicators. 5. The cyber security enthusiast reads the report and decides if there are any indicators of concern. 6. The report yielded malicious indicators.

	7. The cyber security enthusiast decides to analyze network to see if there have been any network connections outward that are suspicious. 8. The cyber security enthusiast proceeds to download the file. 9. The enthusiast saves the report to read and understand it in detail later, to improve his cybersecurity knowledge.
Alternate Flow	1. The report yields a known malicious indicator for the file and a suspicious network connection 2. The cyber security enthusiast decides the software is not safe for installation. 3. The cyber security enthusiast does not save the file onto his/her computer and blacklists the source.
Postconditions	There is no malicious software that is introduced in the system and the cyber security enthusiast learns about malicious indicators and network connections.

Use Case Diagram:

The following is a user case diagram of the use cases that are appropriate with S.M.A.R.T

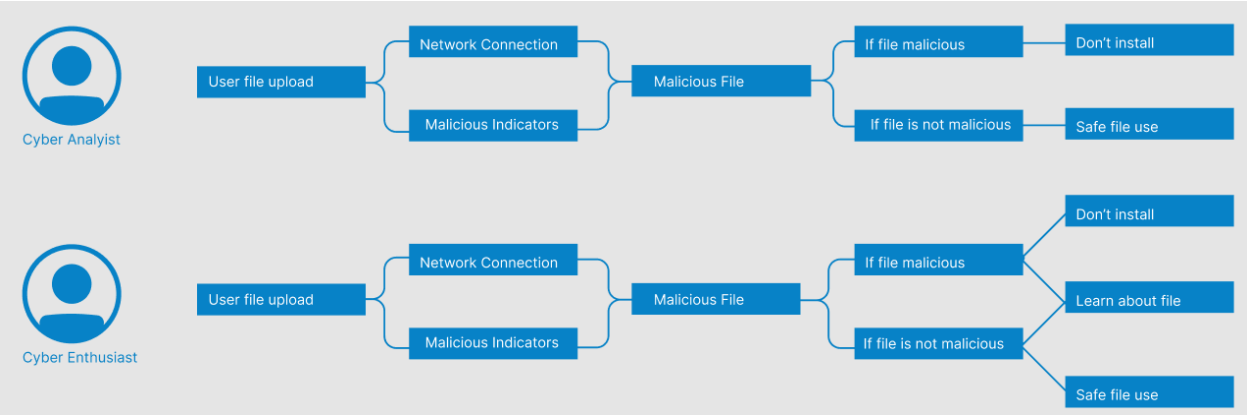


Figure 2:Use Case Diagram

Testing Plan:

Overview

The Testing plan consists of the steps followed by team members to ensure complete functionality of SMART. This section outlines the method used to test each feature, the expected outcome, and the actual result.

Methodology

SMART was tested in weekly and bi-weekly sprints, with Team members meeting to try and evaluate the progress of development. During each meeting, the project was run from scratch to ensure all aspects were integrated together.

Scope

- Does the web-page load?
- Does the page connect the custom CSS applied?
- Is the UI as expected?
- Can a user successfully upload a file?

- Is Cuckoo Sandbox integrating seamlessly?
- Do VirusTotal and PeStudio integrate with the web-page
- Do all tools provide an accurate analysis?
- Is the report visible to the user?
- Can a user successfully download the report?

Objectives

Define what your testing strategy is and what the end goal is.

a. All items under the project scope should be accounted for.
b. The final product must cater to all user personas
c. All use cases must be addressed by the final product
d. All known bugs must be resolved a week before IT Expo

Test Logs and Procedures

Table 7: Test Log and Output Table :

Item #	Test Case#	User and Role	Expected Output	Actual Output	Pass / Fail	Reason	Date
1	N/A – Server running	All roles	Server runs	Server runs	Pass	Works as expected	19th Oct 2021
2	N/A – Web Page loads	All roles	Web page is accessible locally only	Web page is accessible locally only	Pass	Works as expected	19th Oct 2021

3	Multiple instances of Analyses	All Roles	Successful multiple instances running	Running according to the order of input passed.	Pass	Works as expected	1st Feb 2021
4	Simultaneously run multiple analyses in cuckoo sandbox	All roles	Analyses can run simultaneously and concurrently	Instances run sequentially, unavailable to analyze multiple files	Fail	The instance are running on a single VM. Multiple VMs are required for simultaneous execution	3rd Feb
5	Intuitive front-end	All roles	User is able to successfully navigate and upload files	User was able to upload file without external feedback or supervision	Pass	Works as expected	4th Feb
6.	N/A - PeStudio Output to md/txt.	All roles	User is able to successfully get PeStudio is markdown and text format	Successful output	Pass	Works as expected	3rd Feb

Testing Review

The testing process was a success in general, and we believe our plan of meeting for sprints on a weekly basis is successful. Despite being in the initial stages of development, the test results for all our tests were positive, and we believe no changes to the testing plan are required, at this stage.

Change Management Plan:

A change can be requested by any team member. The current work plan involves meeting thrice a week (Tuesday, Thursday, and Saturday), as well as in-person meetings on Mondays. Any member wishing to propose a change can bring it up through Teams or mention it in person. A change will not be approved outside of a planned meeting. In the first meeting following the proposal of the change, all team members will be present and discuss pros and cons for the change, as well as vote on whether or not they are willing to approve it. Team members may be required to do prior research on any technology changes proposed. If team members don't share the same opinions on the change, an external party (the project advisor) may be approached for further feedback. All proposed changes will be either accepted or denied within 3 meetings from the first proposal, or roughly one week. If the change proposed is critical to the project development, this timeline may be shortened, and team members can call emergency meetings to help speed up the process.

Following the approval of a change, a Microsoft teams message will be sent to the Project Advisor notifying them of the change. If the team members believe the change is significant from the original project plan, a formal email may also be sent to the advisor, and a virtual meeting, if needed, may be requested.

Budget:

The budget for this project is a total of $34,054.48 for the first month, with a recurring monthly cost of $26,054.48. The budget includes all software and infrastructure required to recreate this project from scratch. A detailed breakdown of the budget can be found below. Costs are expected to vary as product pricing changes.

Items	Category	Type (Recurring/One-Time)	Cost/Quantity	Quantity	Total Cost
4 x Laptops	Hardware	One-Time	$2,000.00	$4.00	$8,000.00
WiFi	Infrastructure	Monthly	$100.00	$2.00	$200.00
AWS - Linux	Cloud Software	Monthly	$69.12	$1.00	$69.12
AWS - Windows	Cloud Software	Monthly	$135.36	$1.00	$135.36
Microsoft Office 365, Including Teams	Software	Monthly	$12.50	$4.00	$50.00
					$0.00
Labor - Front End (at $40 an hour)	Labor	Monthly	$6,400.00	$2.00	$12,800.00
Labor - Back End (at $40 an hour)	Labor	Monthly	$6,400.00	$2.00	$12,800.00
					$0.00
					$0.00
One-Time Total Cost					**$8,000.00**
Recurring Monthly Total Cost					**$26,054.48**
Recurring Annual Total Cost					**$312,653.76**

Figure 3: Hardware and Software Equipment

Problems Encountered and Analysis of Problems Solved:

One of the biggest issues we had was integrating the front-end and back-end. We had the front-end accepting files, and we had our software (PeStudio) accepting files through the command line. We were unable to pass the user uploaded files to our back-end software and so we reached out to multiple people regarding questions as to how to do this. This included other Senior Design professors, former/current co-op colleagues and even friends and personal contacts. Eventually, we agreed that the best way to do this was using an API to feed user uploaded files to all the software we chose, and then individually scrape the results from each of them before combining data into our UI. While we still haven't identified the ideal way to go about this, we have a better idea on this and intend to work on this through the next semester. Another issue we had was hosting Cuckoo Sandbox. Our current hardware wasn't capable of running a VM and hosting Cuckoo in it due to limited resources. We explored alternative machines and tried installing this in other formats (outside a VM) as well. However, when none of this worked, we reached out to Professor Ryan Moore, who shared an old Server with us, on which we now intend to host Cuckoo Sandbox.

Conclusion

Through the course of the project, we familiarized ourselves with technologies such as Cuckoo sandbox and PeStudio, as well as development software such as Express and Node. Having not used any of these before, the first couple of months were a learning experience, with a lot of time being dedicated to tinkering with the tools. We also learned what resources we had, given that a lot of our problems required reaching out and maximizing our existing resources. In the Spring, we expect to integrate our front-end with our back-end, while also conducting rigorous testing to iron out wrinkles in our front-end. We also hope to have more analysis options for the user, so they can pick whether to conduct just static or dynamic analysis.

References

The State of Breach and Attack Simulation and the Need for Continuous Security Validation: A Study of US and UK Organizations. (2020). *Ponemon Institute*. Retrieved September 20, 2021, from https://l.cymulate.com/hubfs/Whitepaper/Reports/Ponemon%20Report%20The%20State%20of%20Breach%20and%20Attack%20Simulation.pdf?utm_medium=email&_hsmi=100606044&utm_content=100606044&utm_source=hs_automation.

Datta, A., Kumar, K. A., & D, A. (2021). *An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis*. Retrieved September 20, 2021, from https://www.researchgate.net/publication/350886133_An_Emerging_Malware_Analysis_Techniques_and_Tools_A_Comparative_Analysis.