Automated Virus Detection System on Windows Operating Systems using a Batch Script

Eliyas Sala

I. Introduction

Did you know that Windows is the most used operating system? In fact, 76.12% of computer consumers use Windows operating system in 2021 [4]. Since it has existed for such a long time compared to others, hackers and modern terrorists have found loopholes to manipulate computer systems running this operating system. They have created different malwares, like viruses, that are often undetectable by antivirus softwares [2]. The data given above also motivates individuals who seek business opportunities, like myself, to develop a software to secure this system. Although different antivirus softwares exist, most of them are ineffective and lack the consumers' trust. This research proposal briefly discusses how these programs work and their tested performances. Then I propose my unique software-based solution with the different development tools I plan to utilize and an estimated timeline to compelete my first prototype.

II. Related Work

Sukwong et al. examines the most used AV softwares and the algorithmic implementation behind them. He/she first provides the distinction between Signature-based detection vs. Behavior-based detection [5]. Both are ways to detect virus and every AV product doesn't necessarily use the same method to do that. In Signature-based detection, AV product scans a file and assigns a unique indentification to that file. For example, it could use hashing algorithms like MD5 to assign value. Then it evaluates based on patterns of that hashing by comparing it to a remote database containing viral characteristics. If a match exists, then that file is indeed virus infected. Behavior-based detection, another common implementation, analyzes the behaviors of that specific file instead of accuracy matching. It is important to note that there are many more other implmentations that different AV softwares use, but we will focus on these two.

Sukwong et al. continues to discuss and points out the ineffectiveness of AV softwares. Based on an experiment conducted to test the responsiveness of selected AV softwares, their activities were monitored when inentionally given infected files. The softwares in the experiemnt were the following: Avast, Kaspersky, McAfee, Norton, Symantec, and Trend Micro. Avast performed the best by detecting 62.15% of malware. However, it could not complete the scan after that and required a resting time. This clearly shows we shouldn't completely rely on our AV softwares to keep our Windows computers secure.

III. Proposed method

To solve the problem of enhancing virus detection capability on Windows operating systems, it is important to analyze the different types of viruses that exist and learn their behaviors. Based on the statistics provided on the global market share of desktop operating systems, Windows has consistently dominated the market for years in staggering amounts. Other well-known operating systems, like macOS, are far from reaching this success. However, this much of consumer usuage has created vulnerability for Windows operating systems [4]. Unlike most antivirus softwares, my proposed detection software application runs without requiring internet access. It builds upon a Signature-based detection system. And it executes directly from a batch script daily (unlike some AV softwares where users must manually open to troubleshoot their systems), which is connected to Task Scheduler. I am providing an automated scanning software system to run diagnostics on files found from multiple folders and subfolders daily at a specific time. Given a file path to scan for viruses, my software will first use a hashing algorithm to compute the MD5 hashing for every file. Then it matches a database system of text files (recall I said my software works offline) containing large data of the existing viruses and their MD5 classification signitures [3]. After cross-referencing the MD5 generated from scanning the file against our database of known viruses' MD5, my application detects whether is it safe or virus infected. If infected, a batch script should delete the file. By automatically running my program everyday in the background offline, users can be rest assured that at least I won't collect their data and their systems are more secured due to frequent scans.

IV. Future work

Overall goal

Using Microsoft Visual Studio, develop a C/C# or Java based desktop application that is connected to batch

file/files to detect viruses from files without requiring internet access and delete files if diagnosis returns positive.

September 6th – 30th:

I will decide which programming langauge to use based on feasibility. If I choose Java (my most proficient langauge), I can design the user interface using Swing API and create event listeners. I can also import Java's security class to import MessageDigest class to use it and generate MD5 hashing by scanning the files. I might have to use a Scanner object or some sort of BufferReader to scan the files and store them as strings for manipulation. I might also attempt to implement in C# to test if it is better. While I am developing these aspects, I will also start researching on computer viruses' MD5 codes.

October 1st – 31st:

My main goal within this timeframe is to create a database of viruses' MD5 data. I will use a real-world dataset from VirusShare.com. First, I will store the MD5 data in text files and test my application. Then (could be a challenge) I will try to import those text files into an Access database system or mysql to create a database that interacts with my program. But since I want my software to run offline, I might not use a database server. So, I might just use text files as a "database."

November 1st – 30th:

I will create a simple batch script to open my software. This batch file should be connected directly to Task Scheduler to run everyday. Also, another batch script to delete if the scanned files are viruses. To calculate the timing for configuring the deletion batch file on Task Scheduler, I might need to estimate scanning time. Eg: if my program scans a folder and finishes in 30 minutes with some virus-infected files, inside that if statement is where my deletion batch script should be called. But it won't run automatically, and we want it to remove those files as soon as detected. So, I will setup Task Scheduler to run first batch file and deletion batch file within that scanning time gap. I will additionally attempt to create a progression bar to display to users while my program is scanning folders, perhaps add estimate time to finish process. Design status bars: green for safe files and red for virus-infected files after process completes. I must write a logic to loop through folders to search for files most likely in my batch script since I want to automate the system as much as possible. I will develop cool and harmless sample viruses to showcase during my demo. Finally, I will provide a documentation on how to configure my application, inlcuding what times to set up which batch scripts with Task Scheduler.

V. Conclusion

This project proposal explores the challenges faced by computers running Windows operating system when encountering malware. It specifically discusses how computers run antivirus softwares to perform detection operations. Then it provides the ineffectiveness of these AV softwares and supports this claim by citing other related works and experiments conducted by other researchers. It proposes a Signature-based detection method and builds upon that idea to offer alternative detection solutions. Although my proposed sofware application does not fully solve the problem of secuirty against computer viruses, it provides a new approach and unique developmental ideas. This initial prototype will continue to undergo refinement until my desired goals and optimal solutions are achieved.

REFERENCES

[1] Al-Asli, Mohammed et al. “Review of Signature-based Techniques in Antivirus Products.” (2019).

[2] Garba, Faisal A. et al. “Evaluating the State of the Art Antivirus Evasion Tools on Windows and Android Platform.” (2019).

[3] Sahoo, Abhaya, et al. “Signature based Malware Detection for Unstructured Data in Hadoop.” (2014).

[4] “Desktop operating system market share worldwide,” StatCounter Global Stats. [Online]. Available: https://gs.statcounter.com/os-market-share/desktop/worldwide. [Accessed: 05-Sep-2021].

[5] O. Sukwong and H. S. Kim, "Commercial Antivirus Software Effectiveness: An Empirical Study," IEEE Computer Society, pp. 63-70, 2011.