International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

# MALWARE ANALYSIS TOOL

Akshay Shukla, Preeti Naval, Amerendra Kumar

*Department of Computer Science and Engineering*
*B.Tech, Shri Ramswaroop Memorial College of Engineering and Management, Lucknow, Uttar Pradesh*

---

**Abstract -** ***The main purpose of the project is to provide users with security and privacy help and guidance. The scope of the project includes many modules that support users while improving the security of computer systems. Forgery of antivirus software, which is malware disguised as software, is becoming a concern. This Trojan horse method is effective in tricking people into downloading dangerous malware to their systems, thereby exposing them to security risks. You can use this program to increase the security of your system and analyse detected malware. This utility has its own database that guarantees consistency. In general, looking at the security and protection aspects of your system will help your users better understand and access this tool.***

## I. INTRODUCTION

The project is a malware analysis tool with the aim of helping users so that the system used by the person is not disturbed by external threats such as viruses, worms, Trojans and other harmful codes. This tool is a platform independent tool, which means that it can be run on both Windows and Linux, Debian compatible platforms. It offers the user a simple and easy to use GUI so that it can be used all over the world. This tool has different modules, which makes it very different from a normal antivirus program. Because this tool was created in Python, it is easy to use in code. Projects are defined in different ways, so I don't have any users doing any business online, students studying computer-related terms and using different terms on the internet, etc. Users can also use it firmly. Many internets use. The most unique feature of this project is that it provides users with full antivirus functionality as if it were platform-independent, so it can be used in a particular way on any platform on Windows, Mac or Linux systems. In addition, there is a quarantine module that checks all previous malicious files and code to better handle infected files that are said to be dangerous to computer systems. [1]

## II. LITERATURE SURVEY

Malware analysis is a multi-step process providing insight into malware structure and functionality. Behaviour monitoring, an important step in the analysis process, is used to observe malware relations with respect to the system and is achieved by employing dynamic coarse-grained binary-instrumentation on the target system. Initial examination of collected malware is called profiling, Dataflow analysis examines the way data is moved and changed throughout the execution of a program outlined a model where analysis tools are distributed on a local victim machine and on an external machine, to capture behavioural aspects of the malware on the local machine and its interaction with external services over a network. External services as outlined can be setup on the external monitoring segment. A number of analysis tools are utilized by malware forensic analysts, with static and dynamic analysis representing two significant methodologies that can be used to analyse malware. Software disassemblers and debuggers such as IDA Pro (Hex-Rays, 2008) and OllyDBg (Yuschusk, 2008) can be used to perform a detailed analysis of the malware code and provide an internal view of the malware's functionality (Valli & Brand, 2008). This is referred to as static analysis.

- Static Anomaly Detection

Wagner proposed a technique that created a control flow graph (CFG) for a program representing its

system call trace. At execution time this CFG was compared with the system call sequences to check for any violation.

- Hybrid Anomaly Detection

Rabek proposed an anomaly-based technique where static analysis was assisted by dynamic analysis to detect injected, dynamically generated and obfuscated code. Within the program static analysis was used to identify the location of system calls. The programs can be dynamically monitored later to verify that each realistic system call is made from the same location well-known using the static analysis.

## III. OBJECTIVES

The main goal of the project is to provide users with help and support to ensure their security and privacy. The scope of the project is not limited to improving the security of the user's computer system. In fact, your project contains various modules that support your users. We will need to have:

- Signature gathering in anti-malware having a pre-defined repository of static signatures.
- A suitable interface for the system is required that is interactive and completely designed with each type of user in mind.
- It will provide assistance to users so that the system they are using is not affected by external threats like Viruses, Worms, Trojans, and other malicious code.
- This tool will be platform independent
- The data is available to the user and can be updated at will and is fully controlled by the administrator.

## IV. PROBLEM STATEMENT

The project's major purpose is to offer users support and advise in the areas of security and privacy. Several modules are included in the project's scope that assist users and improve the security of their computer systems. Antivirus software forgery, which is malware disguised as software, is a common problem. This Trojan horse method is an excellent way to persuade people to download hazardous spyware onto their computers, leaving them open to security breaches.

## V. EXISTING SYSTEM

Many current systems, such as Yara rules, cuckoo sandbox, GRR and other tools, are available. Analysts employ open-source malware analysis tools to protect themselves from future assaults, predict them, and share their findings. It is common knowledge that malware is a large business, and the quickly spreading malware epidemic will only grow in power and efficiency in the coming years. Getting the crypters, botnets, and zero-days needed for high-level assaults has never been easier thanks to the development of malware trading forums on the dark web. Furthermore, the more complicated the alternatives are, the more difficult they are to comprehend and assess. Analysts can learn more about the attack lifecycle by using open-source malware analysis tools to test, analyse, and document different kinds of harmful activation.[3][4][5].

## VI. PROPOSED METHODOLOGY

The tool's main approach in this project is to fetch a file or a set of files and generate an MD5 hash of the related file. This hash is then used to look for huge files that contain known virus or malware hashes. There are four different modules. There are 4 modules as follows: -

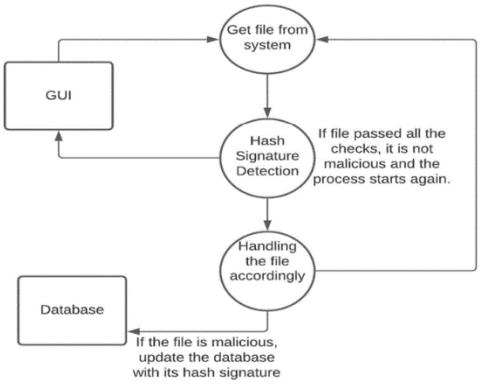
- GUI (GRAPHICAL USER INTERFACE): This is a sort of user interface that allows users to interact with electronic devices using graphical icons and auditory indicators like primary notation, rather than text-based user interfaces, written command labels, or text navigation. It has a simple and user-friendly interface that can be utilised anywhere in the world.

- DATABASE: The signatures of viruses and malware are stored in this database. Because it has its own database that can be updated at

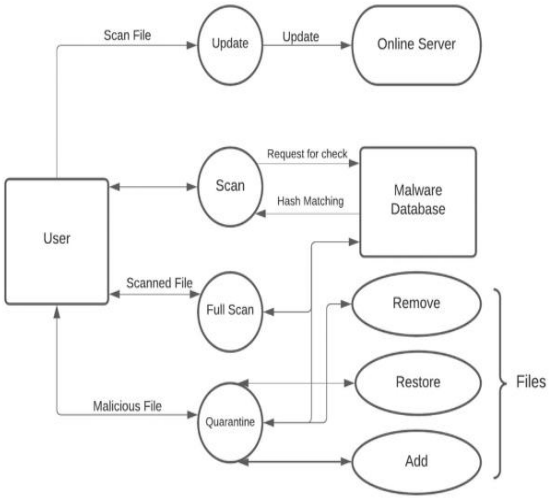
regular intervals, this tool makes dealing with such difficulties much easier.

- DETECTION OF HASH SIGNATURES: The hash is then compared to a database of known viruses and malware.[2][6]

- HANDLING OF THE FILE: The file is handled i.e., if the hash is located the database, then that file is removed/quarantined. If then again, the hash isn't located withinside the database the file is checked and is taken into consideration to be safe.




## VII. BLOCK DIAGRAM




## VIII. IMPLEMENTATION

We will describe the details of the implementations in the work.

- The client starts an output to check for a particular hash from the malware information base.[7]
- Assuming the record is malevolent, it might isolate or erase the document or the scanner might change to another record.[8]
- Full output examines the whole index of records in your PC framework.
- Clients can likewise refresh the mark information base to oblige marks that have been hashed to match possibly malevolent code.

## IX. CONCLUSION AND FUTURE SCOPE

When it comes to fighting malware, the focus is usually on how security software protects your computer from malware. Malware protection (anti-malware and anti-malware in general) is arguably one of the most important defences your computer needs, but what if your computer gets infected before you install the security program? You may encounter a computer that is already infected, especially if you are working in a computer repair or maintenance environment. In such cases, installing and running a security program may not be sufficient to remove all malware. Preventive advice is invaluable, but not very useful if you are using a computer that is already infected. The first step is to download and run the malware scanner.

A lot of things can be improved in this project like:

- A SQL, MySQL, etc database can be added for faster computation.
- The GUI can be improved and made more user friendly with more functionality.

## XI. REFERENCES

[1] Arkajit Datta1, Kakelli Anil Kumar, Aju. An Emerging Malware Analysis Techniques and Tools: A Comparative Analysis (2021)
[2] Yujie Fan, Yan fang Ye, Lifei Chen," Malicious sequential pattern mining for automatic malware detection". Journal in Expert Systems with Applications, (2016).

[3] Sungtaek Oh, Woong Go, Taejin Lee. A Study on the behavior-based Malware Detection Signature (2017)
[4] Ekta Gandotra, Divya Bansal, Sanjeev Sofat. Malware Analysis Tool and Classification: A Survey (2014)
[5] Min Zheng; Mingshen Sun; John C.S. Lui. Droid Analytics: A Signature Based Analytic System to Collect, Extract, Analyze and Associate Android Malware (2013)
[6] Saba Mushtaq, Ajaz H Mir. Signature verification: A study (2013)
[7] Zhao Yong-Xia; Zhen Ge. MD5 Research (2010)
[8] Themis Exarchos, Markos Tsipouras, Costas Papaloukas, Dimitrios I Fotiadis. An optimized sequential pattern matching methodology for sequence classification (2009)