IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 5, November 2024

MidBrain - Antivirus

Omkar S. Kulkarni, Krushna B. Manore, Piyush A. Patil, Aditya B. Jamge
Students, Department of Computer Engineering
Guru Gobind Singh Polytechnic, Nashik, Maharashtra, India

Abstract: *This paper describes the development and implementation of "MidBrain," an antivirus software project to detect and mitigate malware through several techniques, including hash-based malware detection and real-time scanning. The project is student project being pursued towards the completion of a diploma program with the aim of developing rudimental malware detection features, including hash-based scanning and folder scanning with some basic real-time protection. Development of antiviruses is not something which can be practically learnt by experience, but we are students who have accessed any online resources and academic research so that we get into the real picture of antivirus software. We undertake a review of available approaches in antivirus technology, which relates to MidBrain design philosophy in the aim of simpleness, efficiency, as well as protection. Though the scope of this project is small compared to industry leaders like Quick Heal or McAfee, MidBrain would thus serve as a useful learning tool and a foundation for the future expansions.*

Keywords: Antivirus Software, Hash Based Detection, Malware Detection, Real Time Protection, Binary Scanning, Cybersecurity, Malware Prevention, MD5 Algorithm, Antivirus Techniques

I. INTRODUCTION

In the contemporary digital environment, when malware attacks become very widespread, the role of antivirus software is highly important in computer security [1]. However, designing an integrated antivirus package is quite complex and calls for profound knowledge regarding malware behavior and techniques of its detection [2]. The paper presents "MidBrain," a simple antivirus project designed and developed by students as part of their diploma courses. The paper is about a project that comes with some key features, including hash comparison for malware detection [5], scan of a directory, and real-time protection [5]. For the above project, the language used in programming is Python.

The plan for MidBrain is not to be one of the industry players in the league of Quick Heal, McAfee, or AVG but rather to be a training tool to teach students the basics of what needs to be done in an AV implementation [4]. It's practical, and it keeps things grounded in reality; the most basic antivirus will not work on all advanced features or even some of the complicated malware scenarios [5].

Instead, it's an experiential learning experience in the form of a project not complete in all the senses of the word, with detailed learning pathways both via online resources as well as academic researches. As we keep growing, it will add new features assigned based on learning, but for now, it is purely about how the antivirus works.

II. LITERATURE SURVEY

1. Review of Signature-based Techniques In Antivirus Products

It is an overview of signature-based detection methods used in antivirus software. Majorly it considers algorithms like Aho-Corasick and Boyer-Moore to minimize the time complexity in the detection process of malware. The paper describes all the issues in terms of memory usage, time complexity, and limitations with new malwares, suggesting hybrid approaches and heuristics to improve the precision for unknown threats.

2. Heuristic-Based Detection Techniques

In this paper, heuristic-based detection techniques will be used to outline some of the critical contributions toward identification of unknown sophisticated cyber threats. It would so happen that sectors like finance, healthcare, or the government would focus much on such an application. The paper also discusses related challenges such as false positives, resource requirements, and future innovations including quantum-safe algorithms and federated learning toward increased capability in threat detection.

DOI: 10.48175/IJARSCT-22407

3. Behaviour Based Software Theft Detection

This paper reviews the literature on software birthmarks used to identify software theft, with a focus on dynamic methods of program-behaviour-based approaches. In that, it identifies the deficiencies in existing techniques and introduces SCDG - a stronger method of obfuscation and evasion attacks.

4. A Review of Cloud-Based Malware Detection System: Opportunities, Advances, and Challenges

This paper encapsulates the fact that cloud computing has become highly relevant in malware fighting, especially within respect to the protection of IoT and Cyber-Physical Systems. It encompasses a review of various approaches for cloud-based detection strategies that are proposed within this paper and proposes a hybrid framework that combines signature, behavior, deep learning, and heuristic methods. The paper seems to evolve around how cloud-based systems support scalability and increase computational power, which also included certain challenges such as real-time monitoring and obfuscation techniques by advanced malware.

5. Machine Learning Based Intrusion Detection System

In this paper, researchers review the application of machine learning techniques toward developing IDS. Here, it focuses on a comparative study between SVM and Naïve Bayes algorithms using the NSL-KDD dataset. The research study suggests that SVM outperforms Naïve Bayes in terms of accuracy, achieving 97.29% accuracy for intrusion detection purposes; meanwhile, Naïve Bayes performs with a low level of accuracy and increases misclassification rates.

6. SimHash: A Hash-Based Similarity Detection

In this paper, SimHash introduces an efficient similarity file detector which reduces the computational complexity from $O(n^2)$ to O(nlogn). It operates by generating hash keys that map similar files to similar values with the assistance of auxiliary data for refinement. Its technique works well for detecting both relative and absolute similarities but has major weaknesses such as false positives and in terms of scalability. The paper moots future improvements in terms of incorporating metadata and tolerance adjustments to aid in the enhancement of accuracy.

7. Overview of Real-Time Antivirus Scanning Engines

This paper benchmarks several known, open-source and patented real-time antivirus scanning technologies, comparing the strengths and weaknesses of various other different, oddball antivirus systems: stackable file systems (Avfs), Hash-AV virus scanning enhancements, and Dazuko's kernel module that efficiently controls file access. This paper further elaborates on challenges in optimizing between performance and efficiency for real-time scanning, especially with respect to mobile technologies and areas of improvement opportunities.

8. Antivirus Software Versus Malware

This paper discusses evolution, classification, and general analyses of antivirus software and malware. It is developed based on a few definitions of malware by researchers to varied analysis techniques, such as static and dynamic analysis, which are indicated for malware behavior. Other studies on antivirus detection methods, like signature-based, heuristic-based, and behavior- based detection, are considered and their pros and cons considered.

9. Malware in Computer Systems: Problems and Solutions

This paper discusses the literature review of different types of malwares like viruses, worms, spyware, and ransomware like CovidLock; studies all types of such malware impact on computer systems, and looks into countermeasures related to firewalls, antivirus software, and manual techniques for malware removal. Moreover, the study indicates that the University of Halabja does not have adequate protection, and it has suggested training on IT security and licensed software.

DOI: 10.48175/IJARSCT-22407

10. Review of Computer Malware: Detection and Preventive Strategies

Malware discussion on evolution, categories, and obfuscation techniques of malware. Evolution of modern malware refers to its use of self-encryption, polymorphism, and stealth techniques to avoid detection. Prevention strategies discussed include patch management, principles of least privilege, and host hardening against malware.

III. EXISTING SYSTEM

1. Signature Based Detection

One of the simple algorithms by which the antivirus software detects and neutralizes threats is signature-based detection. The process rests upon a unique profile called a signature, which is actually nothing but a pattern of code distinguished by a particular threat [1]. And when it reaches the system, the scanning process goes on within the antivirus software for any existing signatures. If it is unknown, then the software will not recognize it, and therefore, the software cannot prevent it from carrying out its malicious actions. The system creates a signature once it identifies a threat, then add this signature into its database for future reference [1] upon recognizing a threat.

The process is basically reactive; a threat has to first penetrate the system before establishing a signature [1]. For example, if an evil code, named Threat A, deletes files from a computer system, that attack will not even be reported until later, after it has been executed. The system would identify the malicious activity and build up a signature for Threat A. If the attacker attempts to launch Threat A again, then the tool would cross-reference it with its database of the old established known threats and place the entire threat into quarantine.

This method is quite efficient against well-documented malware, since it can quickly identify and neutralize all common threats through the use of established signatures [1]. Besides, it is less resource-intensive as most advanced methods of detection; therefore, one can apply it to a number of systems, for example, which have scarce resources. However, it has serious limitations, especially in the inability to identify new or modified threats that do not match any signature [1][4][5]. For example, if the code of Threat A is changed by replacing an uppercase 'A' by a lowercase 'a', the system may not flag it as a threat, hence enabling the latter to execute its malicious code. The fact that the system base is on past information makes it inherently slow to respond to a new threats; some time has to pass before coming up with new signatures after anattack has occurred.

This means that although the signature-based detection is still very useful in the antivirus arsenal, its limitations make it necessary to add other methods in detection [1] so as to improve protection against this fast changing threat landscape. Notably, signature-based detection does not detect unknown threats, which is a key strength of behavioural detection [3].

2. Heuristic-Based Detection

Heuristic-based detection is a more complex approach used by antivirus software to detect malware by checking the code and behaviour of suspicious programs [1]. Unlike signature-based detection, which compares the in-board virus signatures [1], heuristic analysis recognizes new threats based on their behaviour or some code patterns [1] where the system has no record of the virus and malware. This was developed as a solution against signature-based detection, which is often behind the constantly increasing number of new malware threats. [1]

There are two primary types of heuristic analysis [2]. First one is static heuristic analysis, and second one is dynamic heuristic analysis [2]. In static heuristic analysis, the antivirus software decompiles the suspected program and compares its source code with that of known malicious code stored in a heuristic database. When a piece of code is seen to match one of many predefined patterns, the program is detected as potentially harmful. In dynamic heuristic analysis, it isolates the suspected program and then it is allowed to execute in a controlled environment, or a virtual machine or a sandbox. The antivirus closely monitors the behaviour of the program in that controlled environment. So, if the program somehow shows malicious actions, such as an attempt to delete files or to replicate itself, it will be identified as a virus and blocked [2][5].

The heuristic detection feature detects new and emerging threats independent of signature updates. Still, heuristic detection is not 100 percent perfect. By a static heuristic analysis, It may declare a good program as bad program only if a part of good program matches with the malicious pattern stored in heuristic database, which is false positive.

Furthermore, false negatives can occur because of dynamic heuristic analysis by designing the virus in a way that the malicious actions of the virus occur after it has left the controlled environment. [1][2]

Heuristic detection might be used in combination with signature-based detection or other techniques to ensure full protection. A well-tuned antivirus coordinates these techniques in order to balance the absolutely comprehensive threat detection against minimal errors. The coordination of detecting signatures via both static and dynamic heuristic methods means that the threat detection is always accurate with minimal false positives and false negatives. Detection methods, however, are not faultless. The rate of change in malware demands constant evolution of the antivirus software. [1][2]

Additionally, similar to behavioural detection, heuristic-based detection also faces challenges with false positives, where legitimate programs may be incorrectly labelled as harmful due to certain behaviours or characteristics [3] [5].

3. Behavioural-Based Detection

Behavioural-based detection is a type of detection technique used by antivirus software to scan and analyze how a program behaves as it executes its code rather than relying on existing signatures or analysis of codes [3]. It is a technique more focused on suspect actions or behaviours that would be brought to the front to flag possible malicious activities [3]. The system monitors activities in real time of the program being studied and marks it as malicious if it performs some other previously defined malicious behaviours [3] that are typically linked with malware, such as installation of rootkits, deactivation of security protocols, and generation and execution of files [3].

For example, if any particular program enters the system and starts installing rootkits, the system will recognize this action as dangerous behaviour and will mark the particular program as malicious with the purpose of quarantining it thus it will not be harmful again [3]. However, when an innocuous program arrives in the system, the antivirus will still track its actions, but since it does not have a malicious action, the program will just run without the flag [3].

One of the most important advantages of behavioural-based detection is its ability to detect unknown threats [3], as supported by recent advancements in cloud-based systems [4]. While a given program is not inside the signature database, it may very well be new and even though the antivirus does not know what it is, it may just recognize it as malicious based on how it behaves [3][5]. Detection through this method is never easy [3]. One disadvantage is that it could possibly generate several false positives, where a program that is legitimate is incorrectly labelled as harmful due to the fact that it performs some operations in manners similar to destructive processes [3]. For example, an application may create an error file to log errors, but the act of creating and running files might errantly trigger the operating system to flag the same application as malicious.

This problem raises a question of how effective distinction between injurious and benign behaviour can be made in order to reduce false positives [3]. In a nutshell, behavioural-based detection works by tracking the actions of the program then flagging them up as threats based on the behaviour that the programs are doing [3]. It's very strong protection from new and unknown malware but requires careful tuning in order to minimize the false identification of legitimate programs as threats [3].

4. Cloud Based Detection

Cloud-based antivirus detection is a very new approach towards cybersecurity that has much more advantages with respect to older techniques [4]. In contrast to signature detection, which requires periodic updates for the locally-stored virus signatures [1], cloud-based Antivirus uses the cloud infrastructure for conducting constant, real-time analysis into the threats [4], and hence suspicious files are promptly sent to remote servers in the cloud for instantaneous scanning and thus quickly and more efficiently discover threats [4].

The lone outstanding benefit of cloud-based antivirus software is the way it could identify threats in real time and act on those trends [4]. It achieves this through the principle of cloud computing, where it analyzes the threat at any given time as it strikes. As such, protection deployed becomes much quicker compared to the traditional antivirus software [4]. Additionally, cloud-based solutions do not significantly stress the system resources [2]. Because most computationally intensive work is done on servers, users will feel improved performance of their system and prolonged life for their devices [2].

Another very important feature is scalability [4]. That is, as the needs of businesses grow, cloud-based systems easily readjust without making huge investments in sizeable infrastructures [4]. They also offer centralised management so that IT administrators can easily monitor and control security across multiple devices from one interface [4].

Although cloud-based antivirus solutions have their advantages, they also have some drawbacks [3]. Their effective functioning largely depends on the availability of decent internet connectivity because detection and response are done through the cloud [4]. It's not all as there is a concern of privacy and data security, where sensitive information is relayed to third-party servers for analysis [3], and lastly, latency could creep in on occasions when files have to be sent to remote servers [3] possibly slowing down threat detection which may call for immediate action.

In overall then, cloud-based antivirus is a solution that can abate modern security problems but on the balance of benefits alongside vulnerabilities [4].

5. Machine Learning Based Detection

An advanced method through which modern-day antivirus software detects and puts a halt to the threat is through machine learning-based detection. It is quite the complete opposite of former methods of signature-based detection [1], generated based on predefined virus signatures. This is because machine learning algorithms recognize patterns associated with malicious behaviour [2] in humongous data chunks [2][4] that it analyzes. It gives a set of features extracted from benign files as well as malicious files [3][4] to better achieve the detection capabilities for unknown malware.

This process starts with training the machine learning model on a dataset of labelled examples [2][3], files categorized as either malicious or benign [5]. The model learns distinguishing features between the two categories: specific code patterns, file behaviours, and execution paths [3][4]. After being trained, the model can be used to make real-time predictions about new, unseen files in terms of their potential threat level [3][4] based on the learned patterns [5].

Maybe the biggest advantage developed by machine learning-based detection is flexibility [4]. It will adapt and learn from new data as it continues to be fed [4], thus it will hone its models with more accuracy as time evolves. This responsive method does less relying on constant updates to the database [2] and grants the model the capacity to identify and respond very effectively on newer threats.

Despite obtaining numerous essential benefits out of machine learning, the whole process poses some challenges. For instance, a false positive might occur where legitimate software is flagged as malicious [3][4] and quarantined unnecessarily [5]. Furthermore, the strength of the model relies on the quality and diversity of the training data [3][4]. or complete realization of its potential, the cybersecurity specialists should be constantly checking and updating the algorithm [4] applied in machine learning for robust and proven proof against continually evolving cyber threats.

IV. PROPOSED SYSTEM

The proposed antivirus system, "MidBrain," is primarily supposed to be protection against malware and viruses since it supports some detection techniques that are widely in use. The MidBrain detection system includes hash-based detection, folder and deep scanning, real-time protection, and scanning for a binary to neutralize potential threats. Such a set of techniques forms the heart of functionality in most any antivirus program, so making sure that MidBrain may really scan and combat common types of malwares.

1. Hash – Based Detection

This method uniquely identifies and verifies files by a digital fingerprint; hash plays an important role in the malware detection. During our project we introduced this method. Using the MD5 algorithm, we got a 128-bit hash value that can be compared to known malicious file hashes to identify the malware.

MD5 Algorithm, MD5 also goes by the name of Message-Digest Algorithm 5, meaning it is a cryptographical function which makes a fixed-size hash for any type of input data. As usually represented as a hexadecimal number in length of 32 characters, we can appreciate that with MD5, the rate and promise of consistency will be a great advantage for our antivirus system in generating a consistent hash for the file verification process. [6]

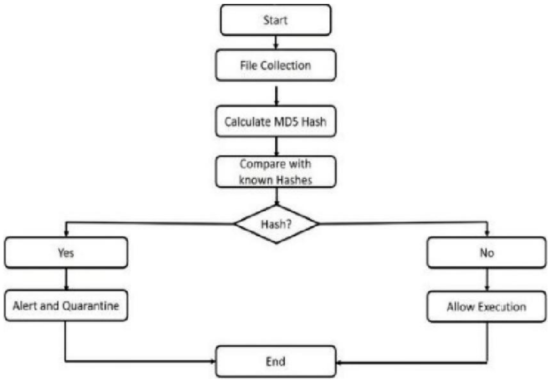

Fig. 1: Flowchart of the Hash-Based Detection Process using the MD5 Algorithm

2. Real Time Protection

Real time protection in the antivirus software continuously monitors the potential threats in real time with continuous vigilance by monitoring files even when they are being created, copied, or modified [1][7]. Unlike the traditional antivirus methods that rely on manual scanning, real-time protection involves continuous scanning without any intervention of the user. This feature detects and neutralizes threats in real time so that malware cannot penetrate into the system.

The main advantage of real-time protection is immediate detection, as soon as malware or undesirable behaviour is encountered—it acts immediately to block or quarantine the threat [2]. Besides file scanning, real-time protection uses behavioural analysis for observing suspicious activities and heuristic analysis for finding new evolving threats that may not exist yet in the virus database with signatures [3][7]. Signature-based detection is also integrated into the system, comparing files to a known database of malware signatures. [7]

When it senses that a threat is present, the antivirus software will trigger an instant reaction to the threat, thereby quarantining the harmful file or effectively killing it so that it cannot harm the computer [4]. This new type of protection devises ensures that any well-known as well as unknown threats will be mitigated before any damage can be caused. In short, real-time protection is designed to ensure constant and automatically secure security when in use.

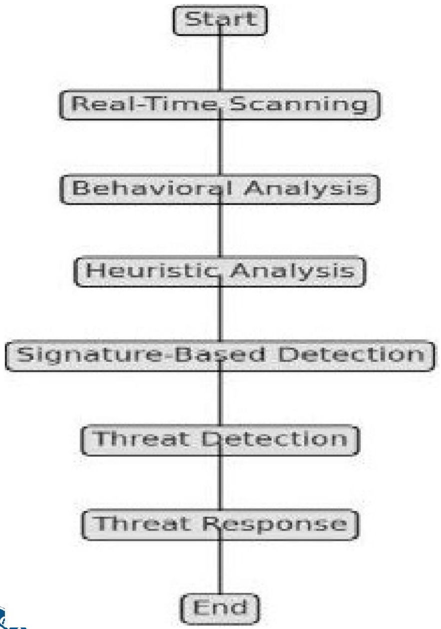

Fig. 2: Flowchart of the Real-Time Protection Process in Antivirus Software

DOI: 10.48175/IJARSCT-22407

3. Binary Scanning

It utilizes hashing for the identification of malicious files, which is a key principle of malware detection. Our antivirus system applies the mechanism of binary scanning by computing the SHA-256 hash for any given file; it then compares this with a database of known malware hashes [6]. For the success of this method, it primarily depends on whether it can match the hash produced by comparison with any of the existing malware hashes because the comparison can generate either a true or false answer. This aids in giving a reliable means to instantaneously identify and report such compromised files without having to run the file.

The significant functions of binary scanning are explained by the following process:

sha256_hash(filename): calculates the SHA-256 hash of the input file and provides a reliable, deterministic way to uniquely determine files.

malware_checker(pathOfFile): evaluates the calculated hash against the malware database for potential matches.

Workflow of binary scanning - The process includes file gathering, which is pulling in the files for scanning. At the system level, it will be calculating hashes, taking a unique hash for each file using the SHA-256 algorithm. At the hash comparison stage, if the hashes match with the malware database and if there is a match, the file is then marked towards malware detection [6].

The process of binary scanning uses libraries in Python to include hashlib to compute hash and os for file operations. The method also uses static analysis, which allows the scanning files without running them; this decreases accidental infection.

Binary scanning has its disadvantages and accelerates to considerable detection of malware with minimized false positives. It has drawbacks, like hash collisions, and two techniques of evasion-obfuscation-and dependencies on known malware hashes [6]. Improvements can be seen in using more than one hashing algorithm and hence, in addition, well-developed methods such as behavioural analysis for unknown or emerging threats.

4. Folder and deep scanning

It involves scanning folders, which is a systematised examination of files and even nested folders in a specified directory. Normally it uses algorithms such as DFS or Depth-First Search and BFS or Breadth-First Search in traversing the directory structure for thorough scanning of all files and subdirectories for potential threats. Deep scanning, on the other hand, scans everything from even the nested directory compressed archives through the more complex technique of scanning with a signature-based technique on one hand and an anomaly-based scanning technique for the detection of known malware as well as irregularities in file behaviour [6].

For better detection, static analysis checks the files but does not run them, while dynamic analysis looks closer at file behaviour when executed. Such strategies ensure achieving balance between accuracy and performance of the system [6]. Techniques like multi-threading (which has the effect of processing as quick as possible), caching scan results to reduce redundant scanning, and exclusion lists for known safe files and directories are all very common optimizations related to scanning folders and deepness. In addition, graph theory and probability theory can be used for modelling directory structures and malware presence likelihood estimation.

Scanning problems include degradation of performance, false positives, and evasions perpetrated by malware to avoid detection. Specialized algorithms are required for the scanning of archives and embedded files within complex file formats. Signature-based scanning uses the concept of file comparison with a database of known malware, whereas anomaly-based scans look out for anomalous behaviour that provides protection against emergent or unknown threats. Further study and development will involve combining machine learning with cloud-based solutions to enhance the accuracy and efficiency of such scanning techniques.

V. FUTURE SCOPE

The present antivirus system implementation is especially important in terms of offering some functionalities such as hash-based detection, real-time protection, folder and deep scanning, and binary scanning. It therefore comes with several scopes for making it more capable in the future.

1. Machine Learning:

The inclusion of machine learning algorithms will enhance the malware detection capability to a great extent. Since hash-based detection may not be able to successfully detect recent patterns of occurrence and anomalies that develop because of a new type of threat, the inclusion of machine learning in such a system may assist in achieving more effective behavioural analysis as well as anomaly detection with the help of machine learning models to identify complex threats.

2. Advanced Heuristic and Behavioural Analysis:

An advanced heuristic and behavioural analysis approach found to develop through this would identify newly developing threats and, in fact even zero-day vulnerabilities, by staying true over time to the behaviour of software.

3. Cloud-Based Threat Intelligence:

This would allow a cloud-based functionality that is constantly updating the malware database of emerging threats to put the system in perpetual vigilance and look out for something new, thus lessened frequency and tedium of manual updates.

4. Mobile Support and Platform Crossing:

Its possible future adaptation can include this making the antivirus mobile and cross-platform supportive with different OS systems like the mobile platforms that specifically run in Android and iOS as well as different OS systems such as Linux and macOS.

5. User Experience Interface Improvements:

Improve the interface so that more visual feedback can be given to the scan results and their personalization settings are available at levels of scanning and security features across multiple levels to enhance the experience of the user.

6. Automatic Update:

The system may incorporate, in the future, an automatic update module ensuring that its antivirus is always up to date with the latest malware signatures and advanced detection algorithms.

Thus, it would be a robust, scalable, and adaptable system against changing cybersecurity threats.

VI. CONCLUSION

The MidBrain antivirus project, in the paper presented, includes malware detection techniques such as hash-based scanning, folder and deep scanning, real-time protection, and binary scanning. The promise is definitely there for a well-designed and planned antivirus solution, albeit still under development. It, however, is designed to use most commonly used techniques like MD5 hashing and SHA-256 binary scanning, thus providing robust protection against known malware threats. And this project is a stepping stone towards providing the next line of improvements for further enhancements and optimizations. This is a preliminary outline of the system; further refinement and testing will then take place in the implementation phase so that MidBrain will become a reliable antivirus tool.

REFERENCES

[1]. M. Al-Asli and T. A. Ghaleb, "Review of Signature-based Techniques in Antivirus Products," in **Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS)**, 2019, pp. 1–6. doi: 10.1109/ICCISci.2019.8716381.

[2]. S. Banik, S. S. M. Dandyala, and S. V. Nadimpalli, "Heuristic-based Detection Techniques," **International Journal of Advanced Engineering Technologies and Innovations**, vol. 1, no. 2, pp. 352, 2022.

[3]. X. Wang, Y.-C. Jhi, S. Zhu, and P. Liu, "Behavior Based Software Theft Detection," in **Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)**, 2009. doi: 10.1145/1653662.1653696.

[4]. Ö. Aslan, M. Ozkan-Okay, and D. Gupta, “A Review of Cloud-Based Malware Detection System: Opportunities, Advances and Challenges,”*EJERS, European Journal of Engineering and Technology Research*, vol. 6, no. 3, pp. 1–8, Mar. 2021.

[5]. Halimaa and K. Sundarakantham, “Machine Learning Based Intrusion Detection System,” in **Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)**, Tirunelveli, India, 2019, pp. 916-920, doi: 10.1109/ICOEI.2019.8862784.

[6]. Sadowski and G. Levin, “SimHash: Hash-based Similarity Detection,” University of California, Santa Cruz, Dec. 13, 2007. [Online]. Available: https://www.webrankinfo.com/dossiers/wp-content/uploads/simhash.pdf.

[7]. L. Radvilavicius, L. Marozas, and A. Cenys, “Overview of Real-Time Antivirus Scanning Engines,” **Journal of Engineering Science and Technology Review**, vol. 5, no. 1, pp. 63-71, Mar. 2012.

[8]. H. Asamoah,“Antivirus Software Versus Malware,” Bachelor’s thesis, Vasyl’ Stus Donetsk National University, Information Technologies Department, Vinnytsia, Ukraine, 2021.

[9]. M. A. H. Saeed, “Malware in Computer Systems: Problems and Solutions,”*IJID (International Journal on Informatics for Development)*, vol. 9, no. 1, pp. 1-8, Jun. 2020, doi: 10.14421/ijid.2020.09101.

[10]. O. K. Akinde, A. O. Ilori, A. O. Afolayan, and O. B. Adewuyi, “Review of Computer Malware: Detection and Preventive Strategies,”*International Journal of Computer Science and Information Security (IJCSIS)*, vol. 19, no. 11, pp. –, Nov. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5847957.