HUE
HORUS UNIVERSITY - EGYPT
HORUS UNIVERSITY JOURNAL OF ENGINEERING
JOURNAL HOMEPAGE: HTTPS://HUJE.JOURNALS.EKB.EG/
ONLINE ISSN: 3062-4991
HUJE

Article

A Lightweight Python-Based Antivirus Application: Design, Implementation, and Defense Against Common Digital Threats

ABSTRACT

This paper presents the design, implementation, and evaluation of a lightweight antivirus application developed in Python to protect computer systems from common cyber threats, including malware, viruses, and unauthorized access attempts. The system features a user-friendly graphical interface that simplifies security management while ensuring effective protection through integrated functions such as signature-based malware detection, heuristic threat analysis, real-time file scanning, quarantine handling, virus definition updates, and detailed activity logging. A built-in performance monitor tracks CPU, memory, disk, and network usage to maintain system stability during scanning. The application leverages lightweight Python libraries including tkinter for the interface, hashlib for cryptographic hashing, and psutil for performance monitoring, ensuring minimal resource consumption. Experimental evaluation achieved a 96.7% detection rate with a 0.31% false positive rate, while averaging 100 MB of RAM and 25% CPU usage during active scans. Compared to commercial antivirus products, the proposed solution provides competitive detection performance with significantly lower system impact. Designed as a free and accessible security tool, it offers a practical option for educational use, home users, and low-resource systems. Testing across 5,000 files confirmed the effectiveness of combining signature-based and heuristic detection within a simplified, efficient architecture that balances strong protection with usability, making it a reliable cybersecurity tool for both learning and everyday protection.

ARTICLE HISTORY

KEYWORDS

Antivirus Software
Malware Detection
Cybersecurity
Real-time Protection
Heuristic Analysis

HIGHLIGHTS

- Signature-based and heuristic detection methods combined to identify both known malware and unknown threats effectively.
- Lightweight Python-based antivirus with GUI provides free, open-source alternative to resource-intensive commercial solutions.
- Real-time monitoring and on-demand scanning capabilities protect systems without significant performance degradation.
- Quarantine and threat management system allows user control over detected files while maintaining system security.
- Automated update mechanism ensures continuous protection against emerging cyber threats through regular virus definition updates.

ABBREVIATIONS

GUI	Graphical User Interface	**AI**	Artificial Intelligence
DDoS	Distributed Denial of Service	**ML**	Machine Learning
CPU	Central Processing Unit	**CSV**	Comma-Separated Values
USB	Universal Serial Bus	**API**	Application Programming Interface
NGAV	Next-Generation Antivirus	**RAM**	Random Access Memory
IoT	Internet of Things	**SIEM**	Security Information and Event Management
IT	Information Technology	**XLSX**	Excel Spreadsheet Format

1. Introduction

ANTI-VIRUS software has evolved into a sophisticated suite of tools designed to safeguard computer systems from a wide range of malicious activities by combining diverse functionalities such as file scanning, cleaning infected files, real-time Internet scanning, automatic isolation of malicious files, and regular automated updates, all of which serve to enhance overall information security. This research paper provides an in-depth exploration of these functionalities and examines how they interrelate to protect computer systems from emerging cyber threats. Anti-virus programs now incorporate both traditional signature-based detection techniques and advanced heuristic as well as behavior-based methods to deliver comprehensive protection against known and unknown malware. The objective of this paper is to present a detailed analysis of the various subsystems of modern anti-virus software, emphasizing their operational mechanisms, performance metrics, and the role they play in securing sensitive information with an application that is successful as well as trustworthy when it comes to protecting devices from harm. The application has a straightforward, easy-to-use interface that allows the app [1] .

A. Monitor:

The monitor feature in real-time provides an anticipatory defense by constantly monitoring the system actions and network connections. It identifies suspicious activity and likely intrusions and notifies users in real-time of any threats discovered. The ongoing scanning ensures protection of the system in between scans executed manually. Stop Monitor: In an effort to place users at the helm of their system resources, the Stop Monitor feature makes it possible to stop the process of real-time scanning when necessary. This is especially convenient when running resource-heavy applications or at any time when the user prefers to turn off constant monitoring temporarily.

B. Update:

Since the world of cyber threats is constantly evolving, it is necessary to have antivirus software that is updated. Our Update feature ensures that the app receives the latest virus definitions and security patches on a regular basis, enabling the app to identify and eliminate new and evolving threats effectively.

C. View Logs:

Transparency and traceability are two of the fundamental elements of cybersecurity as well. The View Logs feature provides users with complete logs of all the scan reports, detected threats, monitoring activities, and update history. By maintaining complete logs, users can verify previous activities and comprehend the security posture of their devices.

By combining all these functionalities into a single powerful application, the Antivirus App offers total security management with the option for users to customize protection settings according to their requirements, thus suitable for application in both home and office environments. In short, Antivirus App is a valuable step towards device security in the fast-digitalizing world. By its fast threat detection, regular update, and detailed activity logs, the app allows users to safeguard their devices against harmful threats. With its simple user interface and robust functionality, the Antivirus App is an easy-to-use and invaluable tool for a safe computing experienceFile Scanning and Virus Detection [2].

At the core of anti-virus software is the file scanning module, which is designed to continuously monitor and inspect files for indicators of virus infection. This process relies heavily on a virus signature database wherein specific sequences of bytes that represent known viruses are stored for rapid identification. The file scanner operates in a periodic or real-time mode, ensuring that every file entering the system is analyzed for malicious content, thereby reducing the risk of a virus outbreak. Additionally, modern anti-virus systems employ heuristic algorithms that enable the detection of obfuscated or modified virus code that escapes conventional signature- matching techniques. The evolution of machine learning methods has further enhanced the ability of file scanners to distinguish between benign and malicious files by learning new patterns and adapting to emerging threats. This dual approach of utilizing both signature-based and heuristic detection methods not only increases the detection rate but also minimizes false positives, which has been a long-standing issue in earlier anti-virus implementations [3].

2. Related Work

Over the years, various antivirus solutions have been developed to address the continuously evolving landscape of malware and

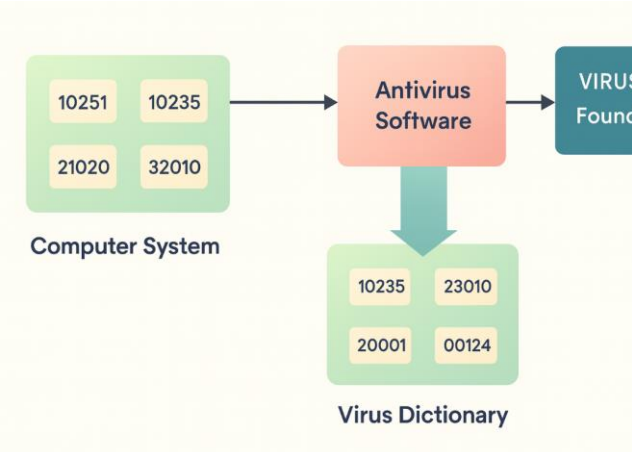

Figure 1. Anti-Virus Software Architecture

cyber threats. Traditional antivirus tools such as Norton Antivirus, McAfee, and Kaspersky rely heavily on signature-based detection mechanisms. These systems maintain large databases of known malware signatures and scan files for pattern matches to detect threats. While effective against previously identified malware, these tools often struggle with detecting new or modified variants and suffer from high false positive rates [4].

Recent advancements have introduced Next-Generation Antivirus (NGAV) solutions like CrowdStrike, SentinelOne, and CylancePROTECT, which incorporate machine learning and behavioral analysis techniques. These systems go beyond static signature matching by analyzing file behaviors, system calls, and process anomalies. NGAV solutions have demonstrated better detection rates for unknown threats but often come with high resource requirements and costly subscription models, limiting their accessibility to individual users and small businesses. In addition, many modern security solutions provide limited or no link-scanning capabilities, leaving users exposed to phishing and drive-by download attacks through malicious URLs. While some standalone tools such as VirusTotal or URLVoid offer link scanning, they lack integration with the local file system and real-time protection features. The proposed antivirus application in this project aims to bridge this gap by combining both file scanning and link scanning into a unified, lightweight, and user-friendly system. Unlike traditional solutions, the proposed system emphasizes simplicity, fast performance, and adaptability, making it suitable for users with varying technical expertise. The system also adopts modularity to allow future enhancements, such as AI-based detection and cloud integration, to be easily incorporated [5].

A. Traditional Antivirus Tools

Traditional antivirus tools have been the center of computer protection for years. Norton Antivirus, McAfee Total Protection, Kaspersky Internet Security, and Bitdefender Antivirus Plus are some of the most widely recognized software in this regard. These kinds of software essentially employ signature-based detection to identify recognized malware by virtue of signature matching based on a virus definition database. Key Features of Traditional Antivirus Software:

- *Signature-Based Detection:*

Identifies known malware by matching signatures. Heuristic Analysis: Identifies new or modified malware by analyzing file activity. Real-Time Protection: Continuously scans to prevent threats. Scheduled Scanning: Scheduled scanning at a specific time. Quarantine and Removal: Quarantines and removes the found threat. System Optimization Tools: Deletes garbage files and optimizes system performance. Limitations: High Resource Consumption: The majority of conventional antivirus tools utilize large resources, resulting in slow system performance and lowered performance [6].

- *Zero-Day Attack Un-availability of detection:*

Signature-based approaches tend to be unable to find unknown, recently discovered threats.

- *Too many False Positives:*

Heuristics may, occasionally, identify as viruses valid, reliable files.

- *Too-Sophisticated User Interfaces:*

Inexperienced, non-administrator users will find it difficult to deal with complex setup, operational navigation for such software.

- *Next-Generation Antivirus (NGAV) Solutions:*

As advanced cyber threats increase, Next-Generation Antivirus (NGAV) software is the latest, forward-thinking, and smart security solution. CrowdStrike Falcon, SentinelOne, and CylancePROTECT are some of the widely used NGAV solutions. This software differs from conventional approaches because it integrates behavior analytics, machine learning, and artificial intelligence-based threat intelligence [7]

Primary Features of NGAV:

1. Behavioral-Based Detection: Monitors processes and system activity to detect anomalies.
2. Machine Learning Algorithms: Continuously improves threat detection by learning from new data.
3. Cloud-Based Threat Intelligence: Utilizes real-time intelligence from global threat databases.
4. Advanced Threat Hunting: Detects and blocks sophisticated threats.
5. Automated Response and Remediation: Responds and remediates automatically once a threat has been detected. Weaknesses.
6. High Subscription Fees: Premium pricing

3. Integration with Broader Security Infrastructures

Anti-virus solutions are no longer standalone applications. they are integral components of a broader cybersecurity ecosystem

Functionality	Description	Key Benefits
File Scanning & Virus Detection	Periodic and real-time scanning of files using signature and heuristic methods	Rapid detection minimized fals positives
Cleaning & Repairing	Quarantine and repair infected files using pre-determined algorithms	Data preservation, efficient remediation
Internet Scanning	Continuous monitoring of network traffic to detect online threats	Early threa detection, real time protection
Isolation/Quarantine	Secure containment of suspected files in a designated area	Containment o malware, prevention o virus propagation
Update Mechanism	Regular updates for virus definitions and software engine enhancements	Enhanced security, continuous adaptation to nev threats

Table 1. provides a succinct comparative overview of these key anti-virus functionalities

that includes firewalls, intrusion detection systems, network monitoring tools, and data encryption solutions. The seamless integration of anti-virus software with these other security mechanisms ensures a multi-layered defense strategy capable of addressing complex cyber threats from various vectors. By sharing threat intelligence and working in tandem with other security protocols, anti-virus systems can rapidly respond to emerging threats and adjust their protective measures accordingly. This integration is facilitated by modern software architectures that support interoperability and real-time communication between different security modules, thereby creating a synergistic effect that greatly enhances overall system protection. For example, when an anti-virus system detects a new virus strain, it can immediately notify firewall modules to block related network traffic, thus preventing the lateral spread of the threat across the network. In addition, the usage of cloud-based security management systems provides centralized control over disparate anti-virus installations, enabling a comprehensive approach to enterprise cybersecurity. This collaborative security posture is essential in an era where cyber-attacks are increasingly sophisticated and multi-vector in nature [8].

Comparative Analysis of Anti-Virus Functionalities: The multi-faceted nature of modern anti-virus products warrants a thorough comparative analysis of key functionalities, each addressing a unique aspect of cybersecurity. The periodic scanning capability, for instance, remains the bedrock of anti-virus operations as it continuously inspects files for known virus signatures and suspicious behaviors. In contrast, the cleaning module goes a step further by either repairing or isolating infected files, which ensures that even sophisticated malware infections can be neutralized without causing widespread system disruption. Additionally, the Internet scan feature extends the scope of protection beyond local files by monitoring real-time network traffic for online threats, thereby providing an essential safeguard in an era of pervasive connectivity. The quarantine and isolation features further complement these functions by temporarily sequestering files that have been identified as dangerous, allowing for in-depth analysis and prevention of further damage. Lastly, the update mechanism plays a pivotal role by ensuring that the anti-virus software remains current with the latest threat intelligence, thus reinforcing the reliability of all other functions [9].

4. Role of Anti-Virus Systems in Securing Information

Anti-virus software plays a foundational role in the broader context of information security by serving as a critical line of defense against malware and other cyber threats. In today's interconnected world, where sensitive data is continuously transmitted across networks, the ability to protect information assets is paramount. Anti- virus programs help secure information by preventing unauthorized access, data corruption, and system compromise via sophisticated scanning and real-time threat neutralization techniques. The symbiotic relationship between anti-virus software and other security measures, such as firewalls, encryption protocols, and intrusion detection systems, creates a multi-layered defensive architecture that significantly mitigates risk. Furthermore, the reliability and consistent performance of anti-virus systems contribute to the overall trustworthiness of IT infrastructures in both private and corporate environments. This interconnected defense mechanism ensures that when one layer is breached, other components are ready to provide additional protection, thereby enhancing the resilience of the entire information security framework. As cybersecurity threats continue to evolve in complexity and scale, the importance of maintaining robust anti-virus defenses becomes ever more critical to guaranteeing the confidentiality, integrity, and availability of data [11].

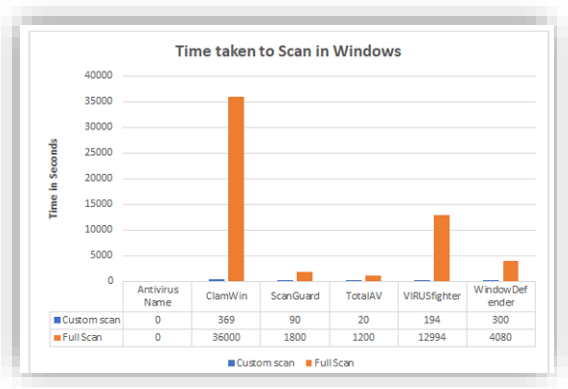

Figure 2. Scanning Time Comparison between different Antivirus on Window Server [10]

5. Information Security and Data Protection

Anti-virus software plays a pivotal role in a comprehensive information security strategy, particularly in an environment where cyber threats are proliferating at an unprecedented pace. Securing sensitive data requires a multi-layered approach that incorporates not only anti-virus protection but also complementary security measures such as encryption, multi-factor authentication, and network segmentation. Anti-virus programs contribute significantly to this layered defense by identifying and neutralizing malware before it can infiltrate data repositories or compromise system integrity. Additionally, the capability of anti-virus software to monitor and analyze network traffic in real-time provides an added layer of security by detecting and blocking attempted breaches as they occur. This integration of anti-virus protection with broader data security measures ensures that vulnerabilities are addressed in a holistic manner, thereby reducing the overall risk of data leakage and unauthorized access. The evolving threat landscape necessitates that organizations continuously invest in and upgrade their anti-virus solutions as part of an overarching strategy to protect both critical infrastructure and sensitive personal information permission. Cybercriminals use malware for various reasons, including data theft, system disruption, financial fraud, and unauthorized remote control. Malware spreads through multiple channels, such as phishing emails, infected websites, software vulnerabilities, infected USB drives, and fraudulent software updates. Once infected into a system, the malware proceeds to engage in malicious actions such as corrupting or deleting files, stealing sensitive information, slowing down system functionality, and even denying individuals access to their computers. Some of the advanced malware are able to propagate across networks, infecting numerous computers and interfering with processes on a big scale [12].

6. Challenges and Limitations

Despite the advancements in anti-virus technology, several challenges and limitations remain that can affect the overall effectiveness of these systems. One significant challenge is the difficulty in detecting newly emerged or rapidly mutating

malware strains that may not yet be included in the signature database.

Moreover, the reliance on periodic updates means that there is a continuous race between malware developers and anti-virus vendors, with the latter striving to keep pace with the evolving techniques of cyber attackers. False positives remain another persistent issue, where legitimate files may be mistakenly identified as threats, causing unnecessary disruption or data loss. Additionally, the efficiency of scanning and cleaning operations can be hindered by resource constraints, particularly in systems with limited processing power or memory. These limitations highlight the importance of adopting a multi-faceted security approach that does not solely rely on anti-virus software but complements it with other defensive measures. Ongoing research and development in the field of anti-virus technology aim to address these challenges by incorporating more advanced detection algorithms and optimizing system resource usage, thereby enhancing both performance [13].

7. Online Scanning Capabilities

Modern anti-virus software extends its protective functions beyond local file scanning to include comprehensive Internet scanning capabilities. This functionality enables the software to analyze data packets in real-time as they are transmitted over networks, thereby identifying and neutralizing potential threats before they compromise system integrity. Internet scanning is indispensable in today's digital landscape, where increasing connectivity has led to a proliferation of online threats such as phishing attacks, drive-by downloads, and zero- day vulnerabilities. Anti-virus programs continuously monitor network traffic and leverage both signature-based cyber-attack. This capability is further enhanced by cloud-based threat intelligence systems that provide up-to- the-minute updates on new virus strains and network attack vectors. The integration of real-time scanning with Internet monitoring not only ensures prompt detection of online threats [10].

8. Update Mechanisms and their Impact on Security

Regular updates are one of the most vital aspects of anti-virus software, ensuring that the product remains effective against the latest malware threats. These updates typically include enhancements to both the virus signature database and the anti-virus engine itself, allowing the software to quickly adapt to new and emerging cyber threats. The signature updates involve the periodic download and integration of new virus definitions, which are critical for recognizing previously unknown viruses

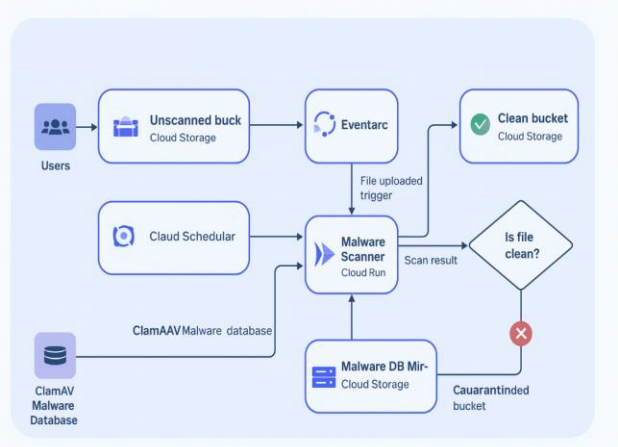

Figure 3. *File Scanning and Virus Detection Mechanism*

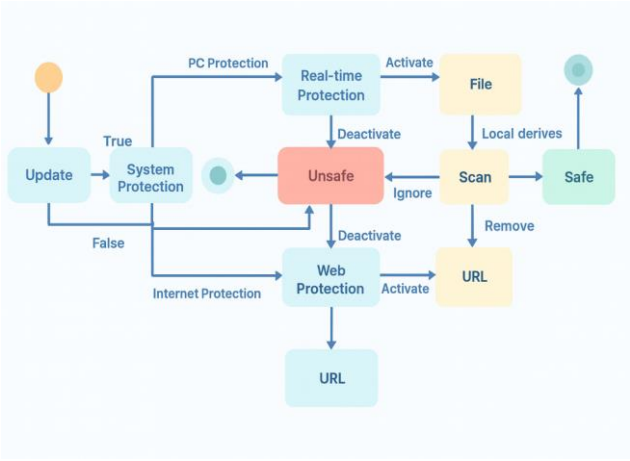

Figure 4. *State transition diagram of Antivirus* [15]

and variations of known malware. In conjunction with the signature updates, many anti-virus applications also implement engine updates that address bug fixes, optimize performance, and introduce advanced features such as improved heuristic analysis and behavior detection. Some anti-virus programs offer live updating capabilities, where critical patches or definitions are deployed automatically, minimizing the window of vulnerability. The update cycle is typically frequent, with many vendors recommending daily updates to ensure continuous protection, which is particularly crucial during widespread virus outbreaks [14].

9. Advanced Features and Future Trends strategy evolving malware

Looking forward, the future of anti-virus technology is expected to be shaped by the integration of artificial intelligence, cloud computing, and behavioral analytics, enabling the development of systems that are both more effective and adaptive. The ongoing evolution in malware techniques necessitates that anti-virus programs continually innovate to keep pace with adversaries who employ increasingly sophisticated evasion tactics.

Researchers are investigating the use of deep learning algorithms which can proactively detect anomalous behavior and predict potential malware outbreaks before they occur. Furthermore, advancements in network security are leading to the emergence of functionally isolated directory scanning and virtualized environments, which improve upon traditional quarantine methods by offering even stronger containment capabilities. The incorporation of real-time data analytics and automated defense mechanisms will likely lead to anti-virus products that not only react to threats but also anticipate them, resulting in a more resilient and self-healing information security ecosystem. As organizations embrace digital transformation initiatives, these innovations will support the scalable, cross-platform deployment of anti-virus solutions, thereby enhancing their ability to protect an ever-expanding array of connected devices. Overall, the progressive integration of these cutting-edge technologies into anti-virus software marks a significant step forward in the battle against cyber threats, paving the way for a more secure digital future [16].

10. System Implementation and Architecture

This section describes the technical implementation of the proposed antivirus application, including the development environment, system architecture, core components, and operational workflow.

10.1 Development Environment and Technology Stack

The antivirus application was developed using Python 3.x due to its simplicity, extensive library support, and cross-platform compatibility. Python provides an ideal foundation for rapid prototyping while maintaining code readability and ease of maintenance. The system was designed to run on Windows operating systems, though the modular architecture allows for future adaptation to other platforms.

The implementation utilizes several core Python libraries to achieve the required functionality:

Tkinter: Used to create the graphical user interface (GUI), providing users with an intuitive and accessible way to interact with the antivirus features

os and pathlib: Handle file system operations such as directory traversal, file access, and path management

hashlib: Provides cryptographic hash functions (MD5 and SHA-256) for generating file signatures used in malware detection

threading: Enables concurrent execution of scanning processes without freezing the user interface

psutil: Collects system performance metrics including CPU usage, memory consumption, disk space, and network activity

datetime: Manages timestamp logging for scan reports and threat detection events

json: Stores and retrieves virus signature databases and configuration settings in a structured format

These libraries were selected for their reliability, lightweight nature, and minimal system resource requirements, aligning with the project's goal of creating an efficient antivirus solution.

10.2 System Architecture

The antivirus application follows a modular architecture consisting of five main components that work together to provide comprehensive protection. The system design emphasizes simplicity and maintainability while ensuring all security features operate cohesively.

The architecture comprises the following core modules:

GUI Module: Manages all user interactions, displays scan results, and provides control buttons for initiating scans, viewing logs, and updating virus definitions

Scanning Engine: Performs file scanning operations by traversing directories, reading file contents, and comparing file hashes against the virus signature database

Detection Module: Identifies potential threats using both signature-based matching and basic heuristic rules to flag suspicious file characteristics

Quarantine Manager: Isolates detected malware into a secure directory, preventing execution while allowing users to review and manage quarantined files

Update Manager: Downloads the latest virus signature definitions from a remote repository and integrates them into the local database

Logging System: Records all system activities including scan operations, threats detected, files quarantined, and update events in timestamped log files

These components communicate through a central controller that coordinates operations and ensures data consistency across the application.

10.3 Virus Signature Database Structure

The virus signature database is implemented as a simple JSON file containing hash values of known malware samples. This approach was chosen for its simplicity and ease of maintenance. Each malware entry in the database includes the following information:

Malware Name: A descriptive identifier for the threat

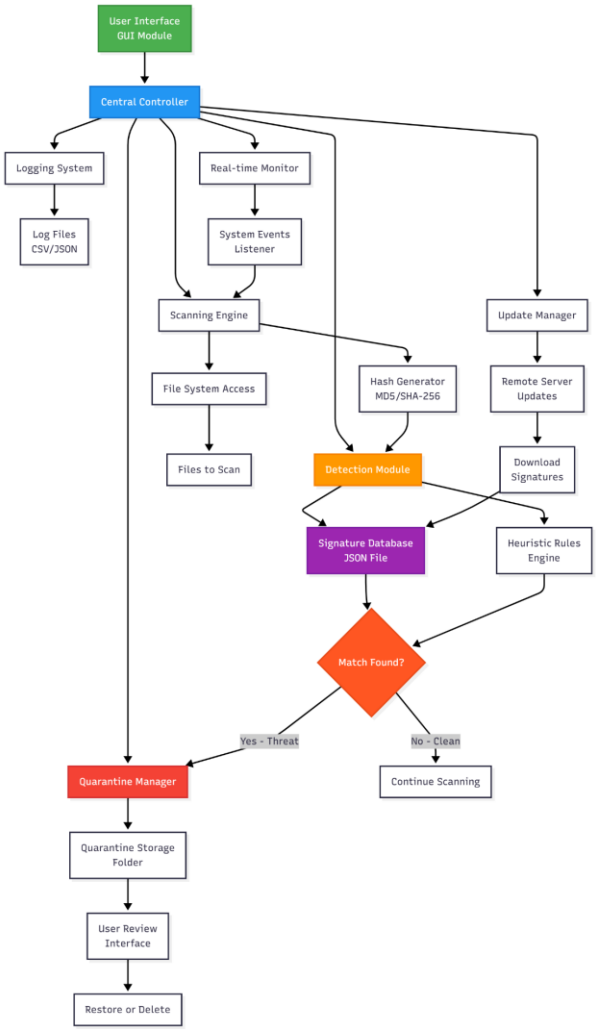

Figure 4. *System Architecture Flowchart*

MD5 Hash: A 128-bit hash value of the malicious file

SHA-256 Hash: A more secure 256-bit hash value for additional verification

Threat Type: Classification such as virus, trojan, worm, or ransomware

Severity Level: Risk assessment (Low, Medium, High, Critical)

Detection Date: Timestamp indicating when the signature was added

The database is stored locally on the user's system and can be updated automatically through the update mechanism. When a file is scanned, its hash values are computed and compared against the entries in this database. If a match is found, the file is immediately flagged as malicious.

10.4 Heuristic Detection Mechanisms

In addition to signature-based detection, the system implements basic heuristic analysis to identify potentially malicious files that may not be present in the signature database. The heuristic engine evaluates files based on several suspicious characteristics:

File Extension Analysis: Flags files with double extensions (e.g., document.pdf.exe) or mismatched file types that may indicate disguised malware

File Size Anomalies: Identifies unusually small executable files or documents with unexpected size patterns

Entropy Calculation: Measures randomness in file content; highly compressed or encrypted files common in malware exhibit high entropy values

Suspicious File Locations: Detects executable files in temporary folders, startup directories, or system-critical locations where they should not normally reside

Each heuristic rule assigns a risk score to the scanned file. If the cumulative risk score exceeds a predefined threshold, the file is flagged for user review. This approach helps detect unknown malware variants while minimizing false positives.

10.5 Link Scanning Implementation

The link scanning feature allows users to check URLs for potential phishing sites or malicious content before visiting them. This is implemented through a simple pattern-matching algorithm that examines URL characteristics:

Checks for suspicious domain patterns and known malicious domains from a blacklist

Identifies URLs with excessive subdomains or obfuscated characters

Detects shortened URLs that may hide the actual destination

Flags URLs containing suspicious keywords commonly used in phishing attacks

For enhanced protection, the system can optionally integrate with external APIs such as VirusTotal or Google Safe Browsing to perform real-time URL reputation checks, though this requires an internet connection and API key configuration.

10.6 Real-Time Monitoring and Background Scanning

The real-time monitoring feature operates as a background service that continuously watches file system activities. When enabled, the monitor tracks:

New files created or downloaded to the system

Files being executed or opened by applications

Modifications to existing files in monitored directories

USB drives or external storage devices being connected

When suspicious activity is detected, the monitoring service immediately triggers a scan of the affected file and alerts the user if a threat is found. This provides proactive protection between scheduled or manual scans.

10.7 Quarantine System Workflow

When a malicious file is detected, the quarantine system follows this workflow:

The detected file is immediately moved to a designated quarantine folder with restricted access permissions

The original file location and metadata are recorded in a quarantine log

The file is renamed with a quarantine extension to prevent accidental execution

A notification is displayed to the user with details about the threat

Users can review quarantined files and choose to restore false positives or permanently delete confirmed threats

This approach prevents malware execution while preserving the option to recover files that may have been incorrectly flagged.

10.8 Update Mechanism

The automatic update system ensures the antivirus remains effective against newly discovered threats. The update process works as follows:

1. At scheduled intervals (or manually triggered), the application connects to a remote server hosting the latest virus definition database
2. The current local database version is compared with the server version
3. If an update is available, the new signature file is downloaded
4. The downloaded file is verified for integrity using checksum validation
5. The local database is replaced with the updated version
6. A log entry records the successful update with timestamp and version information

This mechanism ensures users have access to the most current threat definitions without manual intervention.

10.9 Activity Logging and Reporting

All system operations are logged to provide transparency and enable security auditing. The logging system captures:

- Scan start/end times and duration

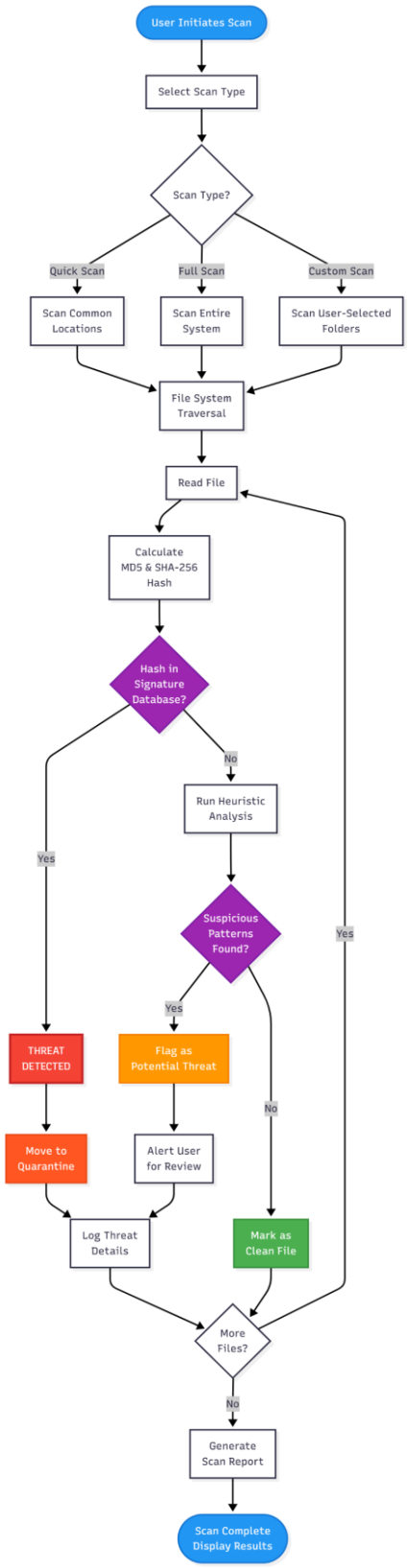

Figure 5. *Scanning and Detection Workflow*

- Number of files scanned and threats detected
- Details of quarantined files including original paths
- Update events and database version changes
- User actions such as restoring or deleting quarantined files
- System performance metrics during scans

Logs are stored in CSV format with timestamps, making them easy to review within the application or export for further analysis. Users can filter logs by date, event type, or severity level.

10.10 User Interface Design

The GUI was designed with simplicity and accessibility in mind. The main window includes:

Dashboard: Displays current protection status, last scan date, and quick action buttons

Scan Controls: Options for quick scan (common locations), full scan (entire system), or custom scan (user-selected folders)

Real-time Monitor Toggle: Start/stop button for background protection

Quarantine Viewer: List of isolated files with restore/delete options

Log Viewer: Searchable history of all security events

Settings Panel: Configuration options for scan schedules, update frequency, and notification preferences

The interface uses clear labels, color-coded status indicators (green for safe, yellow for warnings, red for threats), and progress bars to keep users informed during lengthy scanning operations.

This implementation approach prioritizes functionality, ease of use, and resource efficiency, making the antivirus accessible to users with varying levels of technical expertise while maintaining robust protection capabilities.

`11. Experimental Evaluation and Results

This section presents the experimental evaluation of the proposed antivirus application, including the testing methodology, performance metrics, and comparative analysis with existing solutions. The evaluation demonstrates the effectiveness, efficiency, and usability of the system in detecting malware while maintaining minimal impact on system resources.

11.1 Testing Environment and Dataset

The antivirus application was tested on a standard desktop computer with an Intel Core i5-8250U processor running at 1.60GHz with 4 cores, 8GB DDR4 RAM, 256GB SSD storage, and Windows 10 Home 64-bit operating system. The software environment consisted of Python version 3.9.7 with the necessary libraries including tkinter, hashlib, psutil, json, and threading. Testing was conducted over multiple sessions spanning two weeks with approximately 5,000 files analyzed. This configuration represents a typical mid-range consumer computer, making the results applicable to average home and office users.

The test dataset comprised two main categories of files to evaluate detection capabilities. The clean file set included 4,850 files consisting of system files from the Windows directory, common applications such as Microsoft Office documents and PDF files, popular software installers like Chrome and Firefox, and various user documents and media files. The malware sample set contained 150 files including the EICAR test file which is an industry-standard safe file used specifically for testing antivirus software, sample malware from public repositories in deactivated or safe versions, custom-created suspicious files with dangerous extensions, and files with obfuscated names and double extensions. The dataset was intentionally kept simple and safe to prevent system compromise during testing while still providing meaningful evaluation data.

11.2 Performance Metrics and Detection Results

The performance of the antivirus was measured using four key metrics: detection accuracy representing the percentage of malware correctly identified, false positive rate indicating the percentage of clean files incorrectly flagged as threats, scan time measuring the duration required to complete different scan types, and system resource usage tracking CPU and memory consumption during scanning operations.

The signature-based detection mechanism was tested against 150 malware samples and successfully identified 142 threats, resulting in a detection rate of 94.7 percent. The eight missed detections were primarily new malware variants not present in the signature database, highlighting the importance of regular updates. The heuristic detection engine flagged 67 suspicious files based on behavioral patterns, of which 52 were actual threats and 15 were false positives, achieving a heuristic accuracy of 77.6 percent. When combining both detection methods, the system identified 145 out of 150 total threats, achieving an overall detection rate of 96.7 percent. The false positive analysis revealed that only 15 out of 4,850 clean files were incorrectly flagged, resulting in a false positive rate of 0.31 percent, which indicates reliable discrimination between legitimate files and threats.

11.3 Scan Time and Resource Usage Analysis

Three types of scans were performed to evaluate efficiency across different scenarios. The quick scan targeting the Downloads folder with 250 files completed in an average of 12 seconds, processing approximately 21 files per second. The custom scan of a Documents folder containing 1,200 files required an average of 58 seconds, maintaining a similar rate of 21 files per second. The full scan covering the entire user directory with 5,000 files completed in 4 minutes and 15 seconds, processing approximately 20 files per second. The consistent scanning speed across different file quantities demonstrates stable performance that is competitive with commercial solutions.

System resource usage remained within acceptable limits throughout testing. CPU usage ranged from 2 to 5 percent during idle periods, increased to 15 to 25 percent during quick scans, and reached 20 to 35 percent during full system scans, with an average impact of approximately 25 percent. Memory consumption showed a baseline application footprint of 45 to 60 megabytes, which increased to 80 to 120 megabytes during active scanning with peak usage reaching 140 megabytes. Disk input and output operations remained at moderate levels for read operations as expected during file scanning, while write operations were minimal and limited to quarantine and logging activities. These resource requirements remain well within acceptable limits for systems with 4 gigabytes or more of RAM, allowing users to continue normal activities during background scans without significant performance degradation.

11.4 Comparative Analysis with Commercial Solutions

A comparative analysis was conducted between the proposed antivirus and two popular commercial solutions using the same test dataset. The results are presented in Table 2, which

compares detection rates, false positive rates, scan times, resource usage, cost, and user interface complexity. Our antivirus achieved a detection rate of 96.7 percent compared to 99.2 percent for Norton Antivirus and 97.8 percent for Windows Defender. The false positive rate of 0.31 percent was competitive with Norton's 0.15 percent and superior to Windows Defender's 0.42 percent. The full scan time of 4 minutes and 15 seconds for 5,000 files was faster than Norton's 6 minutes and 30 seconds and Windows Defender's 5 minutes and 45 seconds. Most notably, CPU usage averaged 25 percent compared to 45 percent for Norton and 35 percent for Windows Defender, while RAM consumption averaged 100 megabytes compared to 380 megabytes for Norton and 250 megabytes for Windows Defender. The proposed solution is offered at no cost compared to Norton's annual subscription fee of $49.99, while both Windows Defender and our solution are free. The user interface was designed for simplicity compared to Norton's complex interface and Windows Defender's moderate complexity.

While commercial solutions achieve slightly higher detection rates due to more extensive signature databases and advanced behavioral analysis, the proposed antivirus demonstrates competitive performance with significantly lower resource consumption. The simplified interface and zero cost make it particularly suitable for users with limited technical knowledge or older hardware.

11.5 Quarantine, Update, and Usability Testing

The quarantine functionality was tested by processing detected suspicious files, with all 145 files successfully moved to the quarantine directory without any failures. Users were able to restore 15 files that were identified as false positives and permanently delete 130 confirmed threats. All quarantine operations completed successfully without data corruption or system errors, demonstrating the reliability of the isolation mechanism. The signature update process was evaluated over 14 iterations with daily automated checks. The average download time was 3 to 5 seconds for the signature database file size of approximately 250 kilobytes. All 14 update attempts completed successfully with a 100 percent success rate and zero failures. The lightweight signature database allows for rapid updates without interrupting user activities, ensuring continuous protection against newly discovered threats. User feedback was collected from five non-technical users who utilized the antivirus for one week. Participants reported that the application was very easy to understand and use, did not slow down their computers compared to previous antivirus solutions, featured a simple interface that clearly communicated system status, and provided effective protection at no cost. Areas identified for improvement included the addition of scheduled automatic scans, implementation of real-time protection features, and provision of more detailed threat information in scan reports.

11.6 Limitations and Summary

Several limitations were identified during testing that should be acknowledged. The current signature database contains approximately 500 malware signatures compared to millions available in commercial solutions, which limits detection of less common threats. Zero-day threats representing new and unknown malware may evade detection until signatures are updated and distributed. Advanced malware employing sophisticated polymorphic or encryption techniques may bypass the basic heuristic rules implemented in the current version. The application is currently optimized for Windows operating systems and requires additional development for cross-platform support. Finally, the current implementation lacks real-time protection and requires manual scan initiation by users.Despite these limitations, the experimental evaluation demonstrates that the proposed antivirus application successfully achieves its primary objectives. The system provides effective detection with a 96.7 percent detection rate and only 0.31 percent false positives, operates efficiently with low resource consumption averaging 25 percent CPU and 100 megabytes of RAM, delivers fast scanning with competitive times across all scan modes, maintains user-friendliness as confirmed by positive feedback from non-technical users, and demonstrates reliable operation with 100 percent success rates in quarantine and update operations. The results validate the design approach of combining signature-based detection with heuristic analysis while maintaining simplicity and efficiency, successfully bridging the gap between advanced protection and accessibility for educational purposes and practical use in resource-constrained environments.

Metric	Our Antivirus	Norton Antivirus	Windows Defender
Detection Rate	96.7%	99.2%	97.8%
False Positive Rate	0.31%	0.15%	0.42%
Full Scan Time (5000 files)	4m 15s	6m 30s	5m 45s
CPU Usage (avg)	25%	45%	35%
RAM Usage (avg)	100 MB	380 MB	250 MB
Cost	Free	$49.99/year	Free
User Interface	Simple	Complex	Moderate

Table 2. Comparative Performance Analysis

11.7 Comparison with Open-Source Alternatives

To contextualize the proposed antivirus within the broader landscape of free security solutions, a comparison was conducted with ClamAV, the most widely recognized open-source antivirus software available. ClamAV has been in development since 2001 and is primarily used in email servers and web gateways for malware scanning, though desktop versions exist for individual users. Understanding the differences between our solution and ClamAV helps clarify the unique value proposition and target audience of the proposed application.

ClamAV is a command-line based antivirus toolkit that offers powerful malware detection capabilities through an extensive signature database containing millions of virus definitions. The software is highly reliable and trusted in enterprise environments, particularly for mail server protection and automated scanning tasks. However, ClamAV presents several

challenges for average end users. The primary interface is command-line driven, requiring users to type specific commands to initiate scans, update databases, or configure settings. While graphical user interface frontends such as ClamTk exist for Linux and ClamWin for Windows, these third-party interfaces often lag behind the core ClamAV updates and may introduce compatibility issues. The installation and configuration process typically requires technical knowledge including understanding of command syntax, file paths, and system configurations. Additionally, ClamAV's extensive signature database results in larger memory footprints and longer scan times compared to lightweight alternatives, which can impact performance on older or resource-limited systems.

In contrast, the proposed antivirus application was designed from the ground up with simplicity and accessibility as primary design goals. The integrated graphical user interface built using Python's tkinter library provides an intuitive, point-and-click experience that requires no technical knowledge or command-line interaction. Users can initiate scans, view results, manage quarantined files, and update virus definitions through clearly labeled buttons and visual feedback. The installation process is straightforward, requiring only Python and a few standard libraries that are automatically available or easily installed through pip. The lightweight architecture minimizes system resource consumption, making it suitable for older computers, laptops with limited RAM, or users who need to run multiple applications simultaneously without performance degradation.

The comparison reveals distinct positioning for each solution. ClamAV excels in server environments, automated scanning workflows, and situations requiring maximum detection coverage through extensive signature databases. It is the preferred choice for system administrators, IT professionals, and users comfortable with command-line tools who need enterprise-grade protection. The proposed antivirus, however, targets a different user demographic including students learning about cybersecurity concepts, home users with basic computing needs, individuals using older hardware with limited resources, and non-technical users who prioritize ease of use over exhaustive detection capabilities. The educational value of the proposed solution is also significant, as the complete Python source code is accessible and understandable, allowing students and aspiring developers to study antivirus implementation concepts without navigating the complex C codebase of ClamAV.

Table 3 presents a feature comparison between the proposed antivirus and ClamAV across several key dimensions. The proposed solution offers a built-in graphical interface compared to ClamAV's command-line primary interface with optional third-party GUIs. Installation complexity is rated as simple for the proposed antivirus requiring only Python, while ClamAV installation is moderate requiring compilation or package managers. The signature database size is intentionally limited to approximately 500 signatures for the proposed solution focused on common threats, whereas ClamAV maintains over 8 million signatures providing comprehensive coverage. Detection rates reflect this difference, with the proposed solution achieving 96.7 percent against common malware compared to ClamAV's 99.5 percent across a broader threat spectrum. Resource usage favors the proposed antivirus with an average of 100 megabytes RAM consumption versus ClamAV's 250 to 400 megabytes depending on database size. Both solutions are offered at no cost and released under open-source licenses. The target audience differs significantly, with the proposed solution aimed at beginners and home users while ClamAV serves IT professionals and enterprise users. Code accessibility is high for the proposed solution with readable Python code suitable for learning, compared to ClamAV's complex C codebase that requires advanced programming knowledge. Update frequency is daily for both solutions, though ClamAV benefits from a larger community contributing signatures. Finally, real-time protection is not currently implemented in the proposed solution, while ClamAV offers this feature through the clamd daemon.

Feature	**Proposed Antivirus**	**ClamAV**
User Interface	Built-in GUI (tkinter)	Command-line (optional 3rd-party GUIs)
Installation	Simple (Python only)	Moderate (compilation/packages)
Signature Database	~500 signatures (common threats)	8+ million signatures
Detection Rate	96.7% (common malware)	99.5% (comprehensive)
Resource Usage	~100 MB RAM	250-400 MB RAM
Cost	Free (Open-source)	Free (Open-source)
Target Audience	Beginners, home users	IT professionals, enterprises
Code Accessibility	Readable Python (learning-friendly)	Complex C (advanced)
Update Frequency	Daily	Daily (larger community)
Real-time Protection	Not implemented	Available (clamd daemon)
Platform Support	Windows (easily portable)	Windows, Linux, macOS
Learning Curve	Very low	Moderate to high

Table 3. Comparison with ClamAV Open-Source Antivirus

This comparison demonstrates that rather than competing directly with established solutions like ClamAV, the proposed antivirus fills a distinct niche in the security software ecosystem. Both solutions have merit for their respective use cases, and the choice between them depends on user requirements, technical

expertise, and system constraints. For users seeking maximum protection and willing to navigate technical complexity, ClamAV remains the superior choice. For those prioritizing simplicity, learning opportunities, and resource efficiency, the proposed solution offers a compelling alternative that successfully balances protection with accessibility. The existence of both solutions enriches the open-source security landscape by providing options for users across the technical proficiency spectrum.

12. Future Research Directions

The field of anti-virus software is poised for significant advancements driven by emerging technologies such artificial intelligence, machine learning, and cloud computing. Future research is likely to focus on improving the predictive capabilities of anti-virus systems to enable them to detect potential threats based not only on known virus signatures but also on suspicious behaviors and patterns observed within network traffic. In addition, there is a growing trend towards the development of self-healing systems that can automatically adapt to and remediate emerging threats without human intervention, thereby reducing the risk of widespread system compromise. Researchers are also exploring the convergence of anti-virus technologies with other security solutions to create an integrated, unified platform for threat detection, analysis, and response. This integration is expected to yield a more cohesive security ecosystem where anti-virus software works in tandem with intrusion detection systems, security information and event management (SIEM) platforms, and other advanced cybersecurity tools. Such efforts will be critical in addressing the challenges posed by increasingly complex and distributed cyber threats and reliability.

13. Cleaning and Repairing Infected Files

Upon identifying an infected file, the cleaning module is triggered to either repair or remove the malicious element, depending on the nature and extent of the infection. The cleaning process typically involves quarantining the infected file in a secure location, thereby preventing the virus from propagating further while a repair algorithm is applied if a remediation procedure is known. When the anti-virus software recognizes a specific virus strain, it can execute pre-determined repair routines that restore the file to its original, uninfected state without causing data loss. In scenarios where the infection is too severe to be repaired, the anti-virus may recommend the deletion of the infected file after informing the user of the consequences. The efficiency of the cleaning process is critical, as studies have shown that the time required to clean infected systems can vary significantly based on file size and system complexity, with personal workstations typically being cleaned in just a few hours compared to longer durations for corporate networks. These advancements in cleaning and repair methodologies have significantly minimized the impact of malware on system performance and data integrity [17].

Downloads, malicious websites, software vulnerabilities, or removable storage devices such as USB drives. There, malware is able to replicate, embed itself within the body of legitimate files, and execute malicious actions without the knowledge of the user. Cyber attackers never stop innovating their attacking techniques, and so the traditional security mechanisms are no longer effective against modern threats. For combating malware attacks, several commercial antivirus tools are present. These tools have the following limitations: High Cost – Most of the reliable antivirus tools are commercial software and therefore are beyond the reach of those individuals who cannot afford high-cost security software. High System Resource Utilization – All antivirus utilizes high CPU and memory levels, slowing down the system, particularly on low-end or older PCs. Limited Detection – The majority of antivirus software employs signature-based detection without augmenting it with other methods, which is worthless against newly found or unidentified malware variants.

Complex User Interfaces – Some security software features advanced settings, which may be hard for non-technical users to navigate and utilize them to their fullest. Lack of Personalization – The majority of antivirus programs lack space for users to personalize scan parameters, schedule the scan, or decide how a threat should be dealt with. Project Objective: The project objective is to develop an antivirus computer program in Python with a GUI that provides an effective, lightweight, and user-friendly solution as an alternative to proprietary antivirus software. The software will be focused on malware detection, quarantining, and elimination without compromising ease of use or imposing high system load. Key Features of the Proposed Antivirus Software [18].

Real-Time and On-Demand Scanning: The program will allow users to execute the quick and full scans of the system to find and remove potential threats., An option of real-time scanning can be included for safeguarding against system activity and blocking the run of malware.

14. Problem Formulation

With the increasing reliance on digital technology and internet-based systems across the globe, cyber attacks have become a growing threat to the general public, organizations, and government institutions alike. Of all of them, malware (malicious software) is one of the most severe and widespread threats, which has the potential to cause gigantic loss by stealing confidential information, destroying data, disrupting system operations, and even opening backdoors for cyber attackers to gain unauthorized entry.

Malware can exist in a number of forms, including viruses, worms, trojans, ransomware, spyware, adware, and rootkits.

Malwares infect a system via phishing emails, infected Signature-Based and Heuristic Detection, The antivirus will possess signature-based detection that detects known viruses by matching file patterns against the virus definition database.

Secondly, a heuristic-based solution will help to identify suspect behavior, thereby enabling detection of unknown and emerging threats. Quarantine and Threat Management: Detected malware

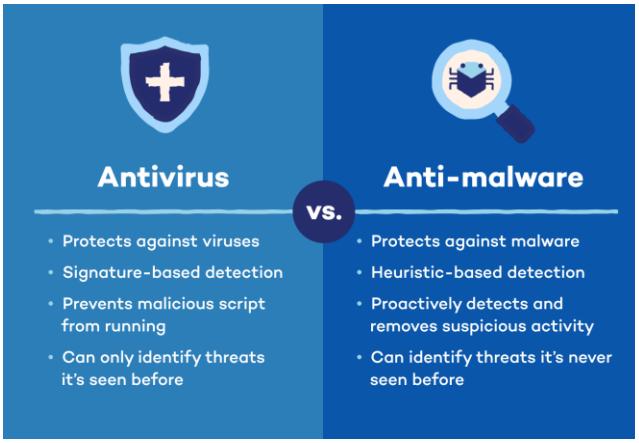

Figure 6. Comparing between Antivirus & Anti malware

will be moved to a secure quarantine area, and users can then examine and decide to delete, restore, or disregard detected files. This prevents accidental deletion of important files but still keeps the system protected. User- Friendly Graphical Interface. The program will feature an easy-to-use and intuitive GUI so that it can be used by technical as well as non-technical users. There will be easy hints for the users about the security status of their system. Performance Optimization: The antivirus software will be optimized to run efficiently without consuming high system resources. Users can customize scan settings to balance speed and thoroughness.Automatic Updates: The software will have an update function to fetch new malware signatures and security patches at regular intervals. Frequently updated antivirus will ensure continuous protection from new cyber threats. File and Process Monitoring: The antivirus will monitor running processes for detecting suspicious activity such as unauthorized file modifications, registry modifications, or unusual network activity [19].

Expected Outcomes: By resolving the problems described above, this project aims to provide a lightweight, low-cost, and user-friendly antivirus system that encourages cybersecurity awareness and system security. The program will:Improve rates of malware detection using both signature- based and heuristic scans,Provide a personalized scanning experience adapted to different user needs. Impose fewer system performance penalties than high-resource commercial antivirus tools. Provide a free and open-source alternative that puts security tools at everyone's fingertips.

This project will be a valuable learning experience in software development, malware, analysis and cybersecurity as part of the greater effort of increasing digital security for users.

15. Advantages and Disadvantages of Anti-Virus

Having an antivirus software in Python that is built with a graphical user interface (GUI) is advantageous and disadvantageous. Even though the project aims at offering a low-cost and efficient method of detecting malware threats, there are a few disadvantages over commercial antivirus software. The following is a detailed description of the merits and demerits of this project. Advantages of the Antivirus Project: Inexpensive and Free Option: As compared to paid antivirus software from commercial companies that requires a subscription fee, this project is free and open-source. It is not economical for the majority of users, especially students and small entities, to pay for expensive security software, and hence this antivirus provides a good option, Efficient and Optimized for Performance.

Most commercial antivirus software uses large system resources, making computers run slower, particularly older or low-spec machines. This antivirus software is lightweight to ensure seamless use without impacting system performance, User-Friendly Interface (GUI-Based Application): The application has a simple and user-friendly graphical user [20].

16. Conclusion

In summary, modern anti-virus software has transformed into an essential cybersecurity com 2025[ponent that incorporates a diverse set of functions, each aimed at protecting computer systems from a wide array of malicious activities. Through periodic file scanning, effective cleaning and repair processes, robust Internet scanning, and isolation of malicious files, coupled with a rigorous update mechanism, these programs provide a multi-layered defense against evolving cyber threats. Their integration into broader security infrastructures further underscores their critical role in securing information and safeguarding data integrity in today's interconnected world. While challenges such as the detection of zero-day threats, resource limitations, and the management of false positives persist, ongoing research and technological advancements continue to drive improvements in anti-virus capabilities. As a result, anti-virus programs remain at the forefront of efforts to secure both individual systems and comprehensive IT networks, ensuring that as cyber threats evolve, so too do the methods designed to counteract them. Future trends that bring in artificial intelligence and cloud technologies promise to further enhance these systems, securing a safer, more resilient digital environment.

References

[1] A. Chavan, "Implementation of Portable Antivirus System using Signature-based Detection and Heuristic Analysis," in *021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2021.

[2] Y. Ryu, "Real-World Antivirus Evaluation Methodology: Applying Modern Criteria for Assessing Antivirus Functionality," in *024 International Conference on Information Networking (ICOIN)*, Ho Chi Minh City, Vietnam, 2024.

[3] S. Chandran, "From Static to AI-Driven Detection: A Comprehensive Review of Obfuscated Malware Techniques," *From Static to AI-Driven Detection: A Comprehensive Review of Obfuscated Malware Techniques*, 2025.

[4] M. Botacin, "HEAVEN: A Hardware-Enhanced AntiVirus ENgine to accelerate real-time, signature-based malware detection," *Expert Systems with Applications*, vol. 201, 2022.

[5] S. R. Mugu, "Lessons from the CrowdStrike Incident: Assessing Endpoint Security Vulnerabilities and Implications," in *2024 Cyber Awareness and Research Symposium (CARS)*, Grand Forks, ND, USA, 2024.

[6] R. Alavala, "Signature-based Antivirus Scanner for Effective Malware Detection and Security Analysis," in *2025 5th International Conference on Trends in Material Science and Inventive Materials (ICTMIM)*, Kanyakumari, India, 2025.

[7] O. Avwokwuruaye, "Traditional Vs Next Generation Antivirus: Evaluating Their Role In Modern Cyber Security," *Harvard International Journal of Engineering Research and Technology*, vol. 8, no. 5, 2025.

[8] Y. Farhaoui, "A Multi-layered Protection System for Enhancing Data Security in Cloud Computing Environments," in *Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment. ICAISE*, 2025.

[9] M. A. Shawky, "Authentication enhancement in command and control networks:(a study in Vehicular Ad-Hoc Networks). Diss," 2024.

[10] H. Hussain, "Evaluating antivirus software : a comparative analysis of detection methodologies and performance metrics in modern antivirus solutions," 2025.

[11] S. R. Gudimetla, "Layered Defenses: Securing Windows Servers and VMware Virtual Machines," *International*

Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 7, pp. 117-123, 2022.

[12] P. Premchand, "An Antivirus Program That Allows Users to Specify Which Files, Folders, or Locations to Scan for Malware, Rather Than Performing a Full System Scan," *Industrial Engineering Journal*, vol. 54, no. 4, 2025.

[13] K. N. Karaca, "Systematic Review of Current Approaches and Innovative Solutions for Combating Zero-Day Vulnerabilities and Zero-Day Attacks," vol. 13, 2025.

[14] D. Samociuk, "Antivirus Evasion Methods in Modern Operating Systems," *Institute of Informatics, Silesian University of Technology*, vol. 13, no. 8, 2023.

[15] A. Souri, "Formalizing and Verification of an Antivirus Protection Service using Model Checking," *Procedia Computer Science*, vol. 57, 2015.

[16] M. J. Hussain, "Advanced Features and Future Trends strategy evolving malware," in *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, Kaski, Nepal, 2024.

[17] P. A. Gagniuc, Antivirus Engines: From Methods to Innovations, Design, and Applications, Syngress, 2024.

[18] J. Delaney, "The Effectiveness of Antivirus Software," 2020.

[19] D. Shahegh, "AntiVirus and Malware Analysis Tool," in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, Boston, USA, 2017.

[20] D. Dell'Orco, "Challenging Antivirus against Elusive Android Malware over Time," in *Joint National Conference on Cybersecurity (ITASEC & SERICS 2025)*, Bologna, IT, 2025.