IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Cyberguard 360: Unified Threat Detection And Protection System

Swapnil Anil Koli, Dineshkumar P ,Prachi Pramod Chavan, Aniket Raghuram Menkudle, Prachi Vinod Pawshe, Anita Parshuram Shinde

Computer Science and Engineering,
Yashoda Technical Campus, Satara, India

Abstract: Cybersecurity threats are become more complex in today's connected world, putting personal and company data at serious danger. In order to overcome these difficulties, CyberGuard360 offers a potent method for identifying malware and other dangers on devices. CyberGuard360 efficiently detects vulnerabilities and malicious activity by using real-time monitoring techniques and sophisticated detection algorithms. This software not only improves security but also gives users useful insights to control and reduce hazards in advance. Users of all technical skill levels may easily navigate and make use of its capabilities thanks to its smooth functionality and easy interface. CyberGuard360 is an important weapon in the continuous battle against cyber dangers because it combines innovation with accessibility, assisting users in maintaining the security and integrity of their digital surroundings.

Index Terms - CyberGuard, Network monitoring, USB device monitoring, Fake email detection ,Malware detection , Real-time system resource monitoring.

I. INSTRUCTION

Strong cybersecurity solutions are more important than ever before in a society that is becoming more digital. CyberGuard360 is an innovative software program made to identify and eliminate malware and threats from your devices. Using the strength of Python and all of its libraries, CyberGuard360 provides a complete solution for protecting your digital environment. CyberGuard360 uses innovative algorithms and real-time scanning capabilities to find vulnerabilities, keep an eye on system activity, and protect your devices from constantly developing cyber threats. It gives consumers peace of mind in a constantly changing environment by empowering them to take charge of their cybersecurity with its user-friendly interface and seamless integration.

II. METHODOLOGY

CyberGuard360 was developed using an organized method to guarantee efficient malware removal and threat identification: Research and Requirement Analysis: Examine pertinent Python libraries, determine user requirements and current cybersecurity concerns. System Design: Construct an architecture with modules for user interfaces, threat detection, and scanning. Development: Use Python to implement the essential features. For detection and analysis, make use of libraries like Scapy, Requests, and Pandas.
Testing: To verify performance and dependability, carry out unit and integration testing while modeling different threat situations. Deployment: Provide full documentation and packaging the application for a simple installation. Monitoring and Updates: Compile user comments and keep threat data and algorithms up to date. User Education and Assistance: Offer materials and assistance to enlighten users on safe and efficient cyber security procedures.

III. FEATURES OF THE CYBURGUARD

The full feature set provided by CyberGuard360 is intended to improve cybersecurity. USB monitoring involves tracking and managing device access, blocking illegal connections, and identifying dangers and threats on the device. Analyzing network traffic is part of network monitoring, which helps spot odd behaviour and possible dangers. Advanced libraries are used in the detection of fraudulent emails and phishing efforts, or "fake email detection". Malware detection uses heuristic and signature-based analysis to find and eliminate malware. Real-time system monitoring ensures that threats are identified right away by continuously examining system activity for abnormalities. Abbreviations and Acronyms

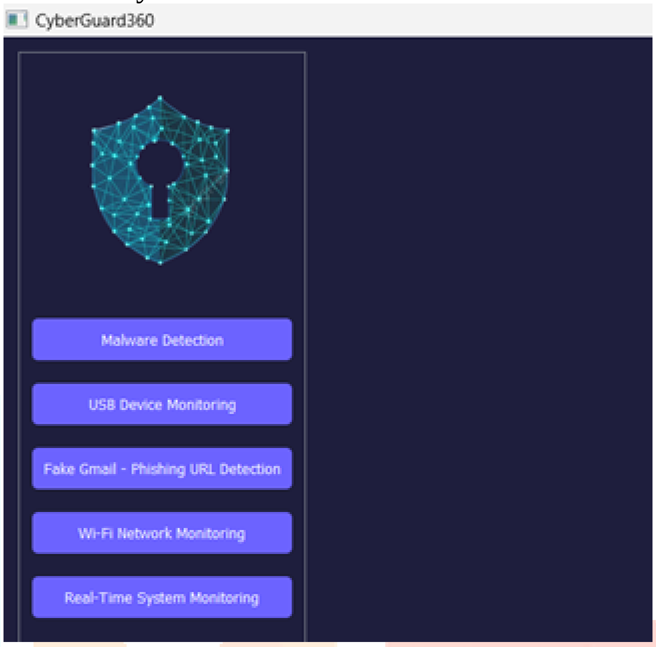


Fig 1 Main GUI

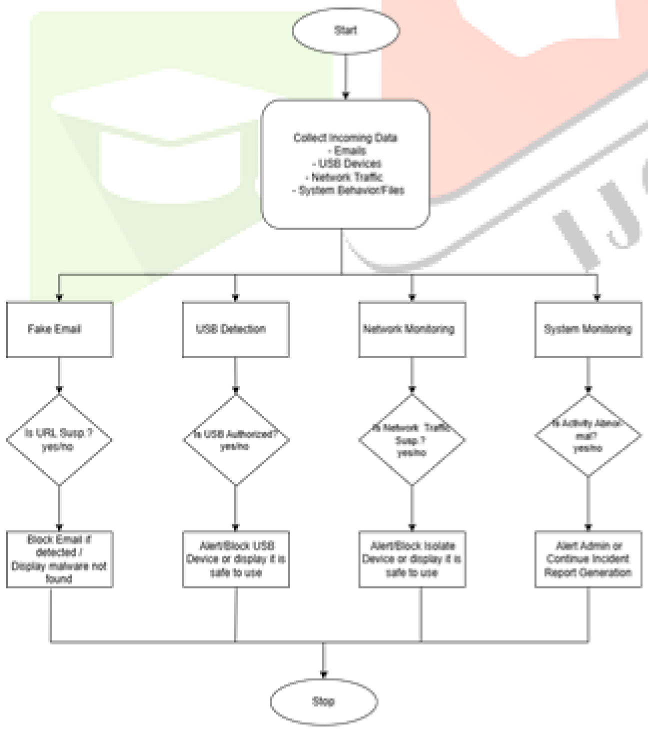
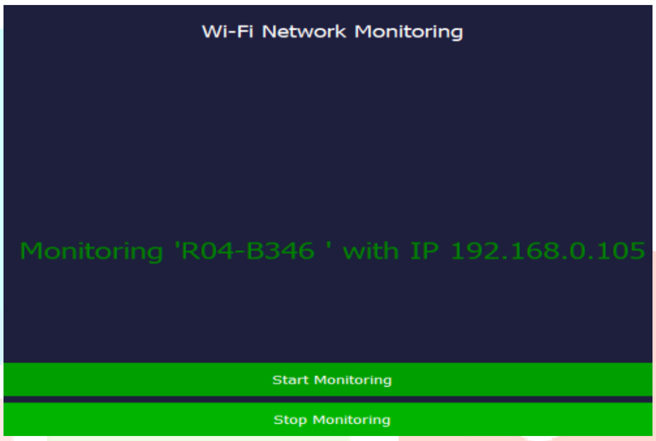

Start
Collect Incoming Data
- Emails
- USB Devices
- Network Traffic
- System Behavior/Files
Fake Email
USB Detection
Network Monitoring
System Monitoring
Is URL Susp.? yes/no
Is USB Authorized? yes/no
Is Network Traffic Susp.? yes/no
Is Activity Abnormal? yes/no
Block Email if detected / Display malware not found
Alert/Block USB Device or display it is safe to use
Alert/Block Isolate Device or display it is safe to use
Alert Admin or Continue Incident Report Generation
Stop

Fig 0.2 Flow Chart

3.1 Network Monitoring System

This application is an intrusion detection system (IDS) designed to use Address Resolution Protocol (ARP) packet analysis to keep an eye out for illegal devices on a Wi-Fi network. This intrusion detection system (IDS), which is intended to identify new or perhaps unauthorized devices on a Wi-Fi network, works by recognizing the distinct MAC addresses of connected devices and recording those that haven't been identified before. By continuously capturing and analyzing packets through threading, the application offers real-time monitoring that can detect possible threats without interfering with network traffic.

Protecting wireless networks has grown essential as network-based dangers have increased, especially in homes and small businesses. Conventional Wi-Fi networks are vulnerable to man-in-the-middle attacks and illegal access. This application is a simple way to use ARP packet analysis to find unfamiliar devices on a Wi-Fi network. An important tool for device identification is ARP, a technique that resolves IP addresses to MAC addresses inside a local network. An effective way to identify illegal network access in real time is using this Wi-Fi Intrusion Detection System. It provides a non-intrusive way to find possibly malicious devices by using efficient threading and ARP packet analysis. For small networks where automatic, low-tech security measures are desired, this approach is perfect. Deeper packet inspection, machine learning integration for behavioral analysis, or warnings for quick response are possible future developments.



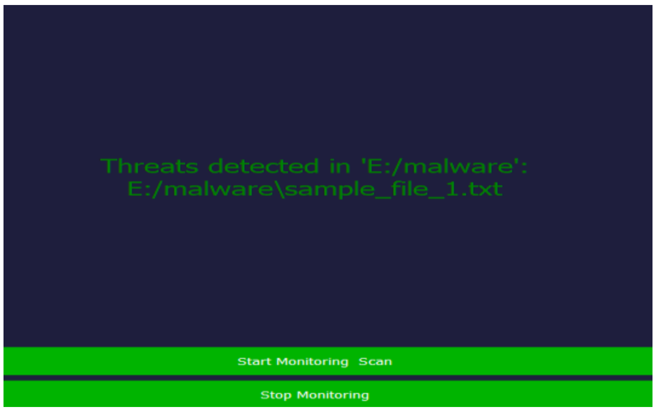
Monitoring 'R04-B346 ' with IP 192.168.0.105

Fig 0.3 Network Monitoring System

3.2 USB Monitoring System

Though USB drives are frequently used for data transfers, they are also a significant source of malware outbreaks. This study presents a method that uses known file hashes to keep an eye out for viruses on USB devices. By identifying and stopping dangerous files from operating on users' PCs, our approach aims to give users an extra degree of security. This study examines current malware detection techniques and describes how this novel strategy uses MD5 file hashing to detect malware better than conventional techniques. By looking for known malware file hashes on connected devices, this study introduces a novel method for detecting malware on USB devices. The system compares USB device files with a predetermined list of virus signatures using the MD5 hashing method. The application dynamically finds USB devices, checks them for potentially dangerous data, and alerts users in real time if any dangers are discovered.

To show how well the system protects user systems from USB-borne risks, the methodology, implementation specifics, and performance outcomes are examined. A collection of known malware hashes was used to compare the system's performance across a range of USB devices. The findings demonstrate that the system can quickly and accurately identify files that match the virus hashes, giving users real-time feedback. The accuracy and detection time were assessed under a variety of situations, including scanning vast amounts of data.



Threats detected in 'E:/malware':
E:/malware\sample_file_1.txt

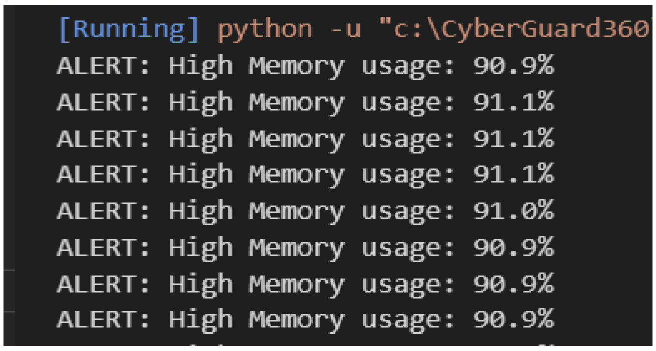
Fig 0.4 USB Monitoring System

3.3 Real Time System Monitoring

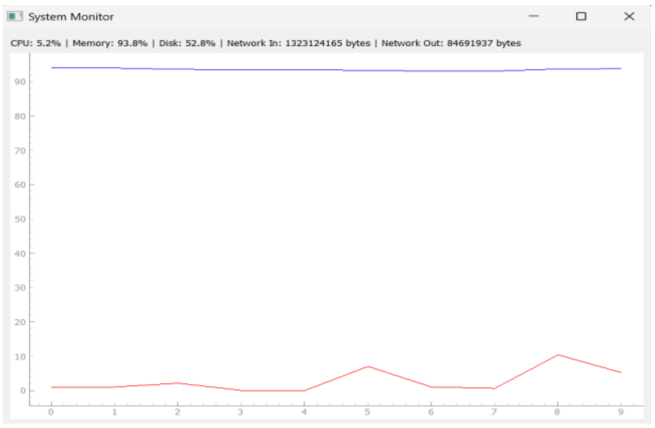
A real-time system monitoring application that monitors a variety of system resources, including CPU, memory, disk space, and network traffic, is presented in this study. The program makes use of Python modules, such as PyQt5 for creating a graphical user interface (GUI) and psutil for gathering system data. In addition to providing real-time resource use updates, the system sends out alarms when specific thresholds are surpassed. The purpose of the monitoring tool is to assist users in effectively managing resource utilization and tracking system performance. With a focus on real-time updates and alert systems, this article examines the implementation specifics, including the software architecture, data collection methodology, and visualization strategies.

The technology makes it easy and efficient to keep an eye on important system resources. The application provides a lightweight solution that is simple to install on any Python-running system by utilizing Python and PyQt5. Users can stay informed about possible problems before they have an impact on system performance thanks to the alert mechanism.

This paper describes a real-time system monitoring application that effectively gives users a configurable approach to keep an eye on system resources. A workable option for proactive system management is provided by the capability to view data in real-time and receive notifications when thresholds are surpassed. The ability to track more system metrics and the inclusion of network consumption graphs are potential future enhancements.



```
[Running] python -u "c:\CyberGuard360
ALERT: High Memory usage: 90.9%
ALERT: High Memory usage: 91.1%
ALERT: High Memory usage: 91.1%
ALERT: High Memory usage: 91.1%
ALERT: High Memory usage: 91.0%
ALERT: High Memory usage: 90.9%
ALERT: High Memory usage: 90.9%
ALERT: High Memory usage: 90.9%
```



CPU: 5.2% | Memory: 93.8% | Disk: 52.8% | Network In: 1323124165 bytes | Network Out: 84691937 bytes

Fig 0.5 Real Time System Monitoring

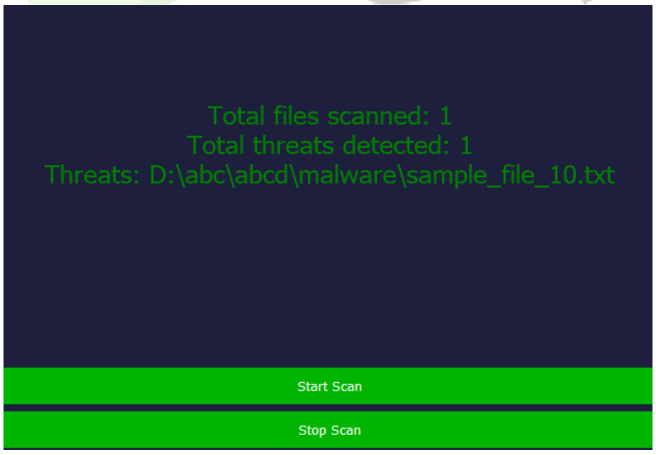
3.4 Malware Detection

Worms, Trojans, ransomware, spyware, and viruses are all considered malware, or malicious software. These programs are made to interfere with, harm, or acquire illegal access to computer systems; they frequently result in system compromise, data loss, or money theft. Malware assaults have the potential to seriously harm a company's finances and reputation. Cybersecurity requires efficient detection techniques to reduce threats and safeguard private data.

1. Overview of Hash-Based Detection: A hash function, such as MD5 or SHA-256, uses the contents of a file to create a unique "fingerprint" for it. A file with even minor changes generates a hash that is entirely different. Hashes for Malware Detection: Security researchers gather known malware hashes. Security systems are able to promptly detect and flag compromised files by comparing file hashes to this database of malicious hashes.

2. Hash-Based Detection's Limitations Insufficient Adaptability: A file's hash will change even if you make just small modifications to it, such updating its information. By making little adjustments to malware files, attackers can avoid detection. Reliance on established samples: The only malware that hash-based detection can detect is known malware. Malware that has been altered or newly discovered cannot be identified unless a hash has been previously recorded.

Threat Intelligence and Crowdsourced Hash Databases: The cybersecurity community may share new malware hashes and receive quick updates thanks to threat intelligence services and community-shared databases.



```
Total files scanned: 1
Total threats detected: 1
Threats: D:\abc\abcd\malware\sample_file_10.txt
```

Fig 0.6 Malware Detection

3.5 Fake Email and Phishing Attack Monitoring

A security system called CyberGuard360 provides a number of tools to assist defend against malware, phishing, and other online dangers. Identifying phony URLs, which are frequently employed in phishing assaults, is one of its primary functions. In order to fool people into disclosing private information, such login passwords or personal information, fake URLs frequently imitate authentic websites. CyberGuard360 employs a number of methods, such as detecting phony URLs, to detect phishing emails. CyberGuard360 can examine the URL of a suspicious or malicious link in an email to see if it fits any patterns of fraudulent URLs.

The goal of CyberGuard360's Fake Mail Detection system, in particular its capacity to identify phony URLs, is to shield users from phishing scams, safeguard private data, maintain organizational security, and lower the likelihood of cyberattacks that could have disastrous outcomes. CyberGuard360 contributes to the development of a safe environment where people and businesses may interact, transact, and browse without worrying about phishing-related dangers by fusing real-time URL analysis, keyword-based detection, reputation checks, AI-powered insights, and user education.

IV. HARDWARE AND SOFTWARE

Hardware Requirements:

Computer/Server: A robust system with adequate processing power (e.g., Intel i5/i7 or AMD equivalent) and RAM (8GB or more) for running the monitoring software.

USB Ports: For detecting and monitoring USB devices.

External Devices: Optional for testing, such as infected USB drives or devices.

Software Requirements:

Operating System: Linux or Windows, depending on your target environment.

Libraries and Frameworks:

Scapy: For network packet manipulation and monitoring.

psutil: To gather system information and monitor running processes.

pyshark: For analyzing network traffic in real-time.

requests: For making HTTP requests to check URLs (for fake mail detection).

smtplib: For sending notifications via email.

Pandas: For fake mail detection to efficiently process, clean, and analyze email data for identifying phishing patterns.

PyDev: For USB detection in Python to monitor and interact with USB devices by leveraging system-level APIs and libraries.

V. CONCLUSION

A CyberGuard 360 is a full solution to today's cybersecurity problems. Organizations can efficiently protect their digital assets by combining advanced threat detection, real-time system monitoring, USB device monitoring, network monitoring, fake email detection, and user-friendly management. Its proactive strategy ensures respect to changing rules while simultaneously reducing risks. Adopting a strong platform like CyberGuard 360 is crucial for preserving security and resilience in today's digital environment, as cyber attacks continue to increase in complexity.

ACKNOWLEDGMENT

We've worked hard on this project. But without the generous assistance and support of my lecturers and fellow group members, it would not have been feasible. I want to express my gratitude to each and every one of them. Yashoda Technical Campus has my sincere gratitude for their leadership, ongoing oversight, provision of project-related information, and assistance in seeing the project through to completion. I want to thank my parents and the staff at Yashoda Technical Campus for their gracious cooperation and support, which enabled me to finish my project. We have been inspired by their dedication to effective inventory control and customer service. We would like to thank everyone who contributed to the creation of this CyberGuard360 once more. This project would not be possible without your help and support.

References

[1] Daniel Sturman a,* , Jaime C. Auton a , Ben W. Morrison b a School of Psychology, The University of Adelaide, Adelaide, SA, Australia b School of Psychological Sciences, Macquarie University, North Ryde, NSW 2109, Australia

[2] Saiful Islam Rimon(B) and Md. Mokammel Haque Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh u1604061@student.cuet.ac.bd, mokammel@cuet.ac.bd

[3] Muhammet Baykara, Ugur Gurturk, Resul Das Firat University, Technology Faculty, Department of Software Engineering, 23119 Elazig, Turkey mbaykara@firat.edu.tr, ugur.gurturk@netix.com.tr, rdas@firat.edu.tr

[4] Safa Rkhouya1 , Khalid Chougdali 2 1 National School of Applied Sciences (ENSA) Ibn Tofail University, Kenitra, Morocco 2 Engineering Science Laboratory, National School of Applied Sciences (ENSA) Ibn Tofail University, Kenitra, Morocco

[5] Jirawat Iamsamang, Natthapon Bijaphala, Pariyakorn Boonperm, Rattapong Lordthong and Phornphop Naiyanetr< Department of Biomedical Engineering, Faculty of Engineering, Mahidol University 999 Phuttamonthon 4 Road, Salaya, Nakhon Pathom, 73170, Thailand Corresponding email: phornphop.nai@mahidol.ac.th

[6] Aparna V Research Intern, Central Scientific Instruments Organisation Council of Scientific and Industrial Research Chennai, India vaparna06@gmail.com

[7] Daniel Sturman a,* , Jaime C. Auton a , Ben W. Morrison b a School of Psychology, The University of Adelaide, Adelaide, SA, Australia b School of Psychological Sciences, Macquarie University, North Ryde, NSW 2109, Australia

[8] Tomoyuki Konnoa,∗ , Anatael Cabrerab, Masaki Ishitsukaa , Masahiro Kuzea , Yasunobu Sakamotoc , for the Double Chooz collaboration

[9] Luigi Gallo a,b, Danilo Gentile a, Saverio Ruggiero b, Alessio Botta b,∗, Giorgio Ventre b

[10] Dongdi Wei, Xiaofeng Qiu Beijing Key Laboratory of Network System Architecture and Convergence Beijing University of Posts and Telecommunications Beijing, China weidongdi@bupt.edu.cn, qiuxiaofeng@bupt.edu.cn