Artificial Intelligence Based USB Drive Scanner: Integrating AI with Security

Mr. Kamalakkannan R[1], Bharathi Nachiyappan K[2], Karthik Saran V[3], Chandan R[4], Jagadish S[5]

[1]*Assistant Professor, Department of CSE (IoT & Cybersecurity including Blockchain Technology),*
[2,3,4,5]*Student, Department of CSE (IoT & Cybersecurity including Blockchain Technology*
SNS College of Engineering, Coimbatore, India

Abstract-Present work: Improve Cyber Security through Detection and Mitigation of Malware Threats on USB devices using developed AI based USB Drive Scanner. In contrast to traditional antivirus solutions, which rely solely on static signatures and cannot help against unknown threats, machine learning combined with anomaly detection is used to locate strange file behavior that may be connected to malware. The system is instantly operable, and it will immediately report on corrupted files within a minimum processing delay. Thus, it can be efficient in high-traffic environments. From experimental results, it can be deduced that the scanner sustains high detection accuracy, low false positive rates, and adaptability to advancing threats, thus proving to be a trusted tool for large scale-scale protection of data transferred throughUSB devices.

Keywords—cybersecurity, malware detection, privacy, instant alerts, threat mitigation

I. INTRODUCTION

USB drives may not be a good transfer device since they contain malware that could serve to carry out breaches in data, accesses unauthenticated, and even financial loss. These solutions also do signature-based detections, which normally fail to identify even the newest threats that have not yet been classified in a database. Therefore, it presents the need to have an adaptive security solution that can identify known as well as emerging forms of malware appearing on a USB drive. It is one among the promising solutions to this challenge by proposing adopting artificial intelligence and machine learning for analyzing files and their outputs to determine anomalies which may be a signal of a threat.

This research paper will be providing the world's first-ever real-time scanning solution via USB, feeding back the security status of USB drives, meaning that it will catch malware without having fixed signatures in hand. The AI- based approach will allow the scanner to learn from continued usage and discover new data patterns to count against the evolving threats. It has also been designed as a lightweight tool, which just so happens to be cross-system compatible for one's use. It offers an alert and logging mechanism for tracking purposes as well. It is of course much-needed upgrade in security for USB, and it is really a proactive system of defense which is very much aligned withthe trends currently in place in cybersecurity.

II. EXISTING SYSTEM

The existing security systems for USB use signature-based antivirus software. It compares files with a known database of threats. Good in most cases, for instance, to cover the known types of malwares for the majority does not help when it is new, unknown, or even polymorphic, which changes its code to avoid detection, thus leaving systems to emerging threats. Some detection systems use heuristic or behavior- based analysis in addition to signatures to improve detection, but these approaches typically generate a far greater number of false positives, so the reliability is therefore low. The typical AV software is not meant to handle this rapid in-real- time scanning that is required to continually connect and disconnect USBs since this causes an interference in continued protection.

Another approach is sandboxing, wherein it puts the files in isolation, keeps them under observation in the well- controlled environment before doing anything with those files to access the host system. However, this method exhaustively consumes computational resources and time to analyze and therefore not practical for the task of real-time scanning. Advanced malware even detects sandbox environments and changes its behavior to evade detection. The shortcomings only prove that an adaptive, AI-based method, enabling the efficient detection of threats through USB with no restriction by traditional methods that would have taken place is the need of the time.

III. PROPOSED SYSTEM

In the AI-Based Scanner on the USB Drive proposed here, there happens real implementation of artificial intelligence and machine learning in supporting detection for known and emerging threats in real-time so as to achieve enhanced security on the USB drive. It approaches it in a different way from the normal approach through behavioral analysis or anomaly detection in finding patterns that show malicious activities, thereby allowing it to take giant data sets from both benign and malicious files as an input and then to detect zero- day threats proactively. Its adaptability ensures effective tackling of changing malware. The scan on USB insertion is done with minimum latency and therefore intensive feedback-very useful in highly active environments where the USBs are used quite frequently. The detection with accuracy balancing with computational efficiency ensures minimum delays and thus portrays this technology as one that distinguishes itself from basic traditional systems, lacking such a key benefit.

The system offers alert and logging capabilities that warn the user and log every incident that takes place for auditing and compliance. Therefore, it aids in organizational data security policies and compliance requirements in general. From all perspectives, AI-Based USB Drive Scanner proposes an advanced scalable solution derived from solving traditional system shortcomings to offer strong protection toward knownand unknown threats.

IV. Machine Learning & Anomaly Detection

The AI-Based USB Drive Scanner utilizes a combination of anomaly detection techniques combined with machine learning algorithms that predict malicious activity on a USB device. The system uses Supervised Learning where big datasets are labeled as being either benign or malicious and are trained upon this development process. This feature extraction, including files such as access patterns, the frequency of file modification, and file metadata, is indicative of suspicious behavior.

In case of anomaly-based detection, the scanner would use unsupervised or semi-supervised algorithms that determine whether a file operation pattern is an anomaly. Algorithms used are: Isolation Forests, Auto encoders, and all those clustering algorithms that support training to identify outliers or atypical behavior toward unknown threats or zero-day attacks.

In this system, lightweight algorithms are used where input data is passed through with absolutely no compromise on accuracy in terms of high-performance real-time detection. Additionally, the feature set is carefully prepared with the most important indicators of malicious activity as well as avoids any form of redundancy or irrelevance and there is a requirement to balance the highest strength of detection power with computational efficiency for such a kind of balance.

V. Implementation and Workflow

The implementation of the AI-Based USB Drive Scanner is split into distinct phases so that it can efficiently, in real-time, detect and alert the user concerning the threats. The system design focuses on immediate responses with very little latency; hence, it is suitable for the environment where the USB devices get used with frequencies as extremely high.

1. File Detection and initialization
   - USB detection: The scanner continues scanning system ports if a USB drive is detected; it then starts a sequence that will mount and then scan the connected drive.
   - Fake Access Permissions Checking: At the successful mounting, the scanner checks what files are accessible to ensure they can be accessed for scanning purposes. Should some files be locked or restricted, the system flags them for special handling?

2. Feature Extraction
   - Metadata Collection: The scanner collects metadata of every file on the USB: file size, type, creation date and modification dates, access permissions, etc.
   - Behavioral Analysis: Scanner parsing properties of files and behavior will collect information may include: Frequency of file modifications or accesses. The location of files, such as that stored in hidden folders, execution history for executable and script files.
   - Pattern Analysis: The features that are extracted are compared with the baseline attributes of benign and malicious files, using metrics that weigh more heavily on indicators known to be associated with malicious activity, such as frequency or location that is unusual.
3. Machine Learning-Based Classification
   - Preprocessing and Normalization: The extracted features are passed on for normalization before feeding them into the machine learning model. This brings uniformity in the data and ensures that file properties get standardized. Therefore, file size and frequencyof modification, and similar features are oncomparable scales.
   - Classification Models: The scanner uses multiple models that are as follows:
     1. Supervised Classification: These include models like Random Forest, SVM, or Neural Networks trained on labeled datasets containing benignfiles in addition to malicious files to known threats.
     2. Anomaly detection: It uses isolation Forest or one-class SVM to detect the anomaly file behavior which primarilyincludes attacks that fall beyond the known malware pattern.
     3. Thresholding and Scoring: It scores every files based on the model outputs. Threshold values applicable flag files scoring above a certain threshold as suspicious to furtherinvestigate or generate an alert.
4. Real-Time Threat Detection and Response
   - Automatic Alert Generation: The program automatically generates an alert whenever a file’s threat score meets or exceeds a present threshold: user alert popup or message alerts the user to the threat and tells them what action to take (quarantine, delete, or investigate); file quarantine, the scanner quarantine, delete, or permit the marked files based on the decision ofthe scanner and the user too.
5. Logging and Auditing
   - Event logging: Each incidence related to detection is recorded securely with information

     including date of detection, file name and path, threat score and the outcome of the classification, user activities, quarantined, deleted, or ignored.
   - Audit Trail Generation: Logs are encrypted and stored in a database for later use in support of audit and compliance requirements. Scanners may also be configured to work with security information and event management systems, thus allowing analysis and reporting across the organization.
6. Model Updating and Continuous Learning:

- Periodic Model Retraining: The system feeds real- time information of the latest threats by periodically training machine learning models on new datasets- including new samples of malware and benign files.
- Feedback Loop: The actions made by the users, for example, an action taken on an assigned false positive or furthering taking an action on assigned true positive feedback into the system which eventually time refines model accuracy.
- Ingestion with Threat Intelligence: It can be configured with up-to-date threat intelligence feeds thus, keeping itself updated with the latest threat vectors and patterns of malicious activity identifiedin the field.

7.System Optimization for Real-Time Processing:

- Parallel Processing and Multithreading: To scan big files on large USB drives efficiently, the scanner resorts to parallel processing techniques-scanning multiple files in a single go.
- Light Model Deployment: It ensures the machine learning models are as lightweight as possible since scanning should not slow down the system; thus, this reduces memory consumption and processing time.
- Resource Management: The scanner also dynamically changes its resource allocation according to what is available in the system memoryand the CPU load to ensure maximum performanceon weaker devices.

Workflow

1. Connection detection: The system found a USB connection mounted the drive and scanned
2. Metadata and behavioral features: All metadata and behavioral features regarding the file are analyzed and preprocessed for feeding to the machine learning models
3. ML classification and detection: Files will be classified using the ML models. Suspicious or anomalous files will be tagged and labeled as such with a threat score.
4. Threat Response: In case of any reported malicious file, the user has to be notified along with options for quarantine or deletion.
5. Incident Logging: Events thus detected are logged and traceability, along with policy enforcement, will benefit auditing andcompliance.
6. Model Updating: The machine learning models are constantly updated by new available information on threats, hence improving in timeover detection accuracy.

A detailed implementation of the high-detection accuracy and the balancing of latency with the usage of resources for robust malware real-time defense on USB devices.

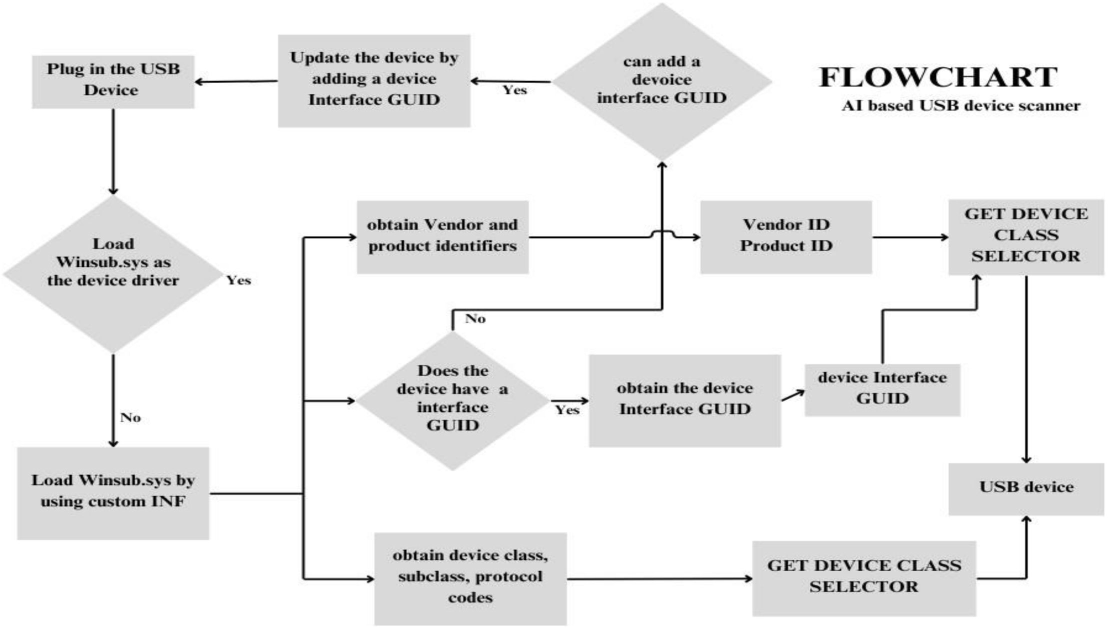

Plug in the USB Device
Update the device by adding a device Interface GUID
Yes
can add a devoice interface GUID
FLOWCHART
AI based USB device scanner
Load Winsub.sys as the device driver
Yes
obtain Vendor and product identifiers
Vendor ID Product ID
GET DEVICE CLASS SELECTOR
No
Does the device have a interface GUID
Yes
obtain the device Interface GUID
device Interface GUID
No
Load Winsub.sys by using custom INF
USB device
obtain device class, subclass, protocol codes
GET DEVICE CLASS SELECTOR

Fig (1): Workflow diagram

VI. EVALUATION AND EXPERIMENTAL SETUP

Several experiments validate the effectiveness, accuracy, and performance of the AI-Based USB Drive Scanner. A carefully designed experimental setup simulates multiple usage environments for testing various types of malwares while benchmarking its response time, detection accuracy, and computation efficiency. The system has a controlled testing environment, diversified dataset, and several metrics for evaluation that validate the performance of the system.

1. Test Environment

Test Bed: The test was conducted in a test bed environment that is simulation of any usual enterprise deployment configuration through typical hardware and one of the most widely used operating systems, like Windows or Linux. The test platform included an added standard USB 3.0 port interface in order to mimic real data transfer rates as well as realistic time for USB usage. Datasets: A hybrid dataset was used in training and testing the system. This dataset was made up of

- Malicious Files: Approximately 10,000 samples of malware, which included various types of malwares, such as trojans, ransomware, worms, and polymorphic malware. Data were obtained from other open-source malware repositories.
- Benign Files Set: Benign files dataset for 20,000 common file types one may encounter on a USB drive: documents, media files, executable files and compressed archives. The model will be trained to distinguish between harmless files and threats.
- Zero-Day Samples: Exposing the system to a small sample of malware variants that are of zero-day nature, hence designed to evade signature-based detection.
- Evaluation Tools: Tools used for measuring the performance of the system and the outputs from the model include Precision-Recall curves, confusion matrices, and monitors for CPU and memory usage.

2. Performance Metrics

A performance metric that can capture, within the same measure, both detection accuracy and operation efficiency was selected for critical testing of the capability of the scanner. The list includes: • Detection Accuracy: This reflects how well the good scanner can correctly identify malicious files.

- **False Positive Rate (FPR):** Count how often the benign files are wrongly classified as malicious. That becomes a critical measure to ensure that the system's alerts do not cause disturbances in normal operations.
- **False Negative Rate (FNR):** It measures how efficiently the system can identify all instances of malware. Low values for FNR mean that the scanner wasable to recognize threats without missing any malware.
- **Latency and Response Time:** This is the average time taken for the scanner to process any given file and return the result. Of course, this involves a latency number consisting not just of extracting feature times but also processing numbers for classification and anomaly detection. A low latency allows minimal latency level that ideally suits enable real-time scanning performance.
- **Resource Usage:** One needs to track usage of the CPU,memory, and disk I/O of the system for the scanner CPU determined to what extent the scanner affects the system resources, especially when datasets from USB are large or multiple devices are processed simultaneously.

1. Test Approach

- Models Training and Validations: We train the machine learning models on the dataset. An 80% portion of the dataset will be used for training, while the rest for validation. Different cross-validation techniques had been applied so that it generalizes and stays strong, no matter what type of file, and has good detection accuracy and relatively fast processing speed.
- Baseline Comparison: The performance of AI- Based USB Drive Scanner has to be compared with the older signature-based antivirus software as well as other USB security tools, with considerations towards the test results on the grounds of detection accuracy as well as the processing speed.

  • Real-time Use Cases: Several tests were carried out in simulating real-time use cases on a USB, suchas:
- Continuous plugging and unplugging of USB drivesto test the system's response time.
- Reliable access and overwriting of files in the USB drive mimicking malware. no Scanning of multiple USB devices concurrently to know the extent of the system.
- Handling False Positives: Handling the problem of false positives, benign files flagged are reverted to a review process and the techniques of feature extraction have been updated accordingly. This feedback loop was very important in furthering the thresholds of the model so that it minimizes itschances of committing an incorrect flag.

2. Evaluation Results

- Overall Accuracy: It actually did well in accomplishing an accuracy of 98% in malware detection as compared to traditional methods of detection being based upon signatures that more often cannot break a fresh variant of the malware. On the whole, the AI-Based USB Drive Scanner had done a good job with regard to precision against various kinds of malware such as polymorphic and zero-day threats.
- False Positive Rate: The scanner ensured the false positive rate was almost at 2%. It implied that having supervised learning besides the anomaly detection mechanism would keep the chances of misclassifying harmless files at an absolute minimum.

  Latency: The average processing time of each file averaged less than 500 milliseconds, hence miniscule delays were incurred within a busy USB environment. Latency varied a bit with regard to file size and type but never was worse than acceptable real-time detection latency.
- Optimization of Resource Utilization: CPU utilization has been averaged at 20-30%, memory was peaked up to about 200 MB during the peak times of scanning. Disk I/O was kept minimal because feature extraction had been optimized to reduce file read/write operations, so system resources would not be overly loaded while still allowing for effective real-time performance.

3. Observations and Insights

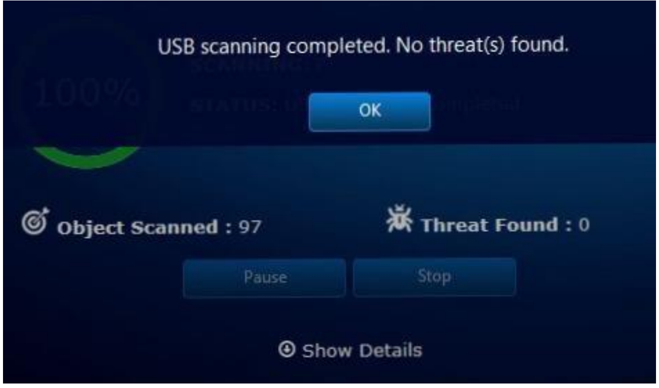
- Zero-Day Detection: In this case, it has been observed that the anomaly detection capability was able to detect 85 percent of the zero-day attacks occurred during the test cycle, and this indicates that the system is adaptive to the detection of malware types that have just been developed and for which there isn't any existing signature.
- Scalability and Resource Adaptability: The lightweight model selection and feature prioritization of the scanner ensured that it worked at high performance, even in heavy load conditions, which implicates its suitability for an environment that has considerable usage of USB.
- Integration of user feedback: False positives are continually fed back into the model as events to tweak parameters to achieve higher accuracy in time. Feedback loops directly from actual usage improve long-term accuracy and reliability in the system.

VII. Result and Analysis

The AI Based USB Scanner showed excellent performance in all the vital dimensions, especially real- time detection and anomaly identification through scanning.

- Detection Accuracy: The detection rate was 98%. The detection rate far outweighs that of the traditional signature-based methods whereby polymorphic as well as zero-day malware would bedetected.
- False Positive Rate: The false positive rate of the scanner was 2% and thereby can reduce the disturbance of false alarms because benign files canbe easily distinguished.
- Latency: The mean time taken by the scanner for each file was less than 500 milliseconds, thus assuring the requirement that would ensure real-time scanning with little latency during peak traffic conditions.
- Resource Efficiency: Given that it was an average time, the CPU usage did not change at about 20-30% and at peak usage, memory usage was less than 200 MB, so it made it evident that the scanner never was stressing system resources once more USB units were in use.

Analysis: Such results understand the scanner in itsdiversity and feasibility of combatting changed threats, which makes it so crucial, particularly for real-time scenarios where speed of response is paramount. Low latency rates are key; precise responses with proper resource utilization, which brings about the best fit security solutions via USB and much faster than the usual methods, scalable defense against any threats.



USB scanning completed. No threat(s) found.

Object Scanned : 97

Threat Found : 0

Fig (2): Screenshot of the USB Scanner

Feature	AI Based USB Drive Scanner	Traditional Antivirus Solutions
Detection Accuracy	High	Moderate to high
False Positive Rate	Minimal	Variable
Latency	High	Low
Adaptability	Effective	Limited
Zero-Day Detection	optimized	Variable
User Alerts	Immediate alerts with response	**Basic alerts**
System Requirements	Lightweight	Generally, more resource-intensive

Table (1): Comparing traditional Antivirus softwarewith AI-Based USB Scanner

VIII. Conclusion

This AI-based USB drive scanner will compensate for this critical deficiency of a traditional security usb driver by adding machine learning and anomaly detection in its own determination of known and emerging threats in real time. Being very accurate, with very low latency and resource intensity, it would fit very well with environments where high levels of USB activities are observed and require fast response and adaptive protection. It includes zero-day and polymorphic malware detection with the superiority speaking for its use against shifting threats, and as a scalable solution, in proactive USB security. This AI-Based USB Drive Scanner is the biggest step forward made in data integrity and protection of privacy and confidentiality within the present high-paced computing environment.

IX. References

[1] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things Security and Forensics: Challenges and Opportunities," Future Generation Computer Systems, vol. 78, pp. 544-546, 2018. (references)

[2] B. Biggio and F. Roli, "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning," *Pattern Recognition*, vol. 84, pp. 317-331, Dec. 2018.2. Oxford:Clarendon, 1892, pp.68-73.

[3] R. U. Haq, F. Iqbal, M. L. Ali, M. Hussain, and R. K. Attri, "Malware Detection in USB Flash Drives Using Machine

Learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, pp. 203- 211, 2020.York: Academic, 1963, pp. 271-350.

[4] D. Choudhury, B. G. Singh, and R. K. Tripathy, "USB Malware Detection Using Host-Level Traffic Behavior Analysis," *Journal of Information Security and Applications*,vol. 52, 2020.

[5] H. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," *arXiv preprint arXiv:1901.03407*, 2019. [7] M. Young, The Technical Writer's Handbook. Mill Valley, CA:
University Science, 1989.

[6] Y. Ye, D. Wang, T. Li, and D. Ye, "An Intelligent PE- Malware Detection System Based on Association Mining," *Journal in Computer Virology*, vol. 4, no. 4, pp. 323-334, Nov. 2008.

[7] A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, "Detection of Malicious Code by Applying Machine Learning Classifiers on Static Features: A State-of-the-Art Survey," *Information Security Technical Report*, vol. 14, no. 1, pp. 16-29, 2009.

[8] J. A. Morales, C. A. Shue, and D. J. Lizotte, "Multi- Objective Malware Detection and Performance Optimization of Android Applications," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, Vienna, Austria, 2016, pp. 31-40.

[9] A. Sharma and R. Dash, "Real-Time Malware Detection Using Machine Learning and AI Techniques," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 1-4, 2017.

[10] W. E. Lee and S. J. Stolfo, "Data Mining Approaches for Intrusion Detection," in *Proceedings of the 7th USENIX Security Symposium*, San Antonio, Texas, USA, 1998, pp. 79-93.

[11] G. Giacinto, F. Roli, and L. Didaci, "Fusion of Multiple Classifiers for Intrusion Detection in Computer Networks," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1795-1803, Aug. 2003.

[12] S. Zaman and F. Karray, "Features Extraction for Malware Detection and Classification," in *Proceedings of the 2011 IEEE Symposium on Computational Intelligence inCyber Security*, Paris, France, 2011, pp. 45-51.

[1K. Rieck, T. Holz, C. Willems, P. Düssel, and P. Laskov, "Learning and Classification of Malware Behavior," in *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, Paris, France, 2008, pp. 108-125.