myresearchgo

myresearchgo Volume 2, Issue 7, July 2026, ISSN: 3107-3816 (Online)

SentinelUSB: Detection of Unauthorized USB Device Usage Using Windows Registry, Event Log Analysis, and Real-Time Monitoring

Kairunnisa
Department of Information Technology
Maharashtra College of Arts, Science and Commerce
Mumbai, India
Kairunnisabilali7@gmail.com

Dr. Saima Shaikh
Head, Department of Information Technology
Maharashtra College of Arts, Science and Commerce
Mumbai, India

Prof. Arun Shaikh
Head, Department of Computer Science
Maharashtra College of Arts, Science and Commerce
Mumbai, India

Abstract—The rapid adoption of portable USB storage devices has significantly improved data portability and operational convenience. However, these devices have also become one of the major sources of cybersecurity threats, including malware propagation, unauthorized data transfer, insider attacks, and digital evidence tampering. Traditional endpoint security solutions primarily focus on antivirus protection and network-based defenses but often provide limited visibility into unauthorized USB device usage. This research presents SentinelUSB, a Windows-based cybersecurity and digital forensics framework designed to detect, monitor, and analyze unauthorized USB device activity through historical forensic analysis and real-time monitoring.

The proposed system integrates Windows Registry (USBSTOR) analysis, Windows Event Log auditing, and Windows Management Instrumentation (WMI) to identify previously connected devices, monitor new USB insertions in real time, and enforce configurable security policies using dynamic whitelisting. A web-based dashboard provides centralized monitoring, forensic reporting, incident logging, and whitelist management for administrators. Experimental observations demonstrate that SentinelUSB successfully detects USB connection events with minimal system overhead while providing comprehensive forensic evidence for security investigations. The proposed framework offers an effective and lightweight solution for educational institutions, enterprises, and digital forensic laboratories seeking improved endpoint security and automated USB device monitoring. Future enhancements include artificial intelligence-based anomaly detection, cloud synchronization, cross-platform compatibility, and enterprise-level endpoint integration.

Keywords—Cybersecurity, Digital Forensics, USB Device Monitoring, Windows Registry, Windows Event Logs, Windows Management Instrumentation (WMI), Endpoint Security, USBSTOR, Whitelisting, Python.

I. INTRODUCTION

Universal Serial Bus (USB) storage devices have become one of the most widely used methods for transferring digital information due to their portability, affordability, and high storage capacity. Although these devices provide considerable convenience, they also introduce serious cybersecurity challenges. Unauthorized USB devices can be used to steal confidential information, introduce malicious software, bypass

organizational security policies, and compromise critical systems. As cyber threats continue to evolve, organizations require efficient mechanisms for monitoring removable storage devices and preventing unauthorized access to sensitive information.

Traditional endpoint protection systems generally rely on antivirus software, firewall configurations, or enterprise Data Loss Prevention (DLP) solutions. While these technologies offer significant protection, they often require complex deployment, centralized management infrastructure, and high licensing costs. Small organizations, educational institutions, and research laboratories frequently lack such resources, making lightweight and cost-effective security solutions increasingly important.

The emergence of Digital Forensics has further highlighted the importance of collecting historical evidence related to USB device activity. Windows operating systems maintain valuable forensic artifacts within the Windows Registry and Event Logs that record information about previously connected USB devices. These artifacts include Vendor IDs, Product IDs, Serial Numbers, timestamps, and device installation history. Proper analysis of these records enables investigators to reconstruct user activity and identify unauthorized hardware usage.

To address these challenges, this research proposes SentinelUSB, a Windows-based USB security monitoring and forensic analysis framework. The system integrates Registry Forensics, Event Log Analysis, Windows Management Instrumentation (WMI), dynamic whitelist verification, and real-time security monitoring into a unified platform. Unlike conventional endpoint protection solutions, SentinelUSB combines historical forensic evidence with live USB monitoring while maintaining minimal resource utilization.

The proposed system is developed using Python and native Windows APIs, enabling rapid deployment without requiring expensive enterprise infrastructure. Through its integrated web dashboard, administrators can monitor USB activity, manage authorized devices, generate forensic reports, review security incidents, and configure security policies from a centralized interface. This research demonstrates how combining forensic analysis with real-time monitoring significantly improves endpoint security while supporting digital forensic investigations.

II. LITERATURE REVIEW

Cybersecurity researchers have extensively investigated USB device security, endpoint protection, and digital forensic analysis. Windows Registry artifacts have long been recognized as valuable sources of forensic evidence because they preserve detailed records of connected USB storage devices. Studies have shown that the USBSTOR registry location contains device identifiers, serial numbers, vendor information, and timestamps that assist investigators in reconstructing historical USB usage.

Several researchers have also explored the forensic significance of Windows Event Logs. Event Logs record driver installation events, Plug-and-Play activities, storage mounting operations, and other system events that provide chronological evidence during cyber investigations. These logs improve the accuracy of forensic reconstruction by validating Registry findings and identifying device connection timelines.

Microsoft introduced Windows Management Instrumentation (WMI) as an administrative framework capable of monitoring hardware and software events in real time. Numerous security applications utilize WMI to detect Plug-and-Play events, automate system management tasks, and monitor hardware changes without requiring kernel-level modifications. Real-time monitoring significantly reduces the response time required to detect unauthorized USB insertions.

Commercial endpoint security products such as Data Loss Prevention (DLP) systems provide device control capabilities but often involve expensive licensing, centralized servers, and considerable computational overhead. Many existing solutions prioritize either real-time monitoring or historical forensic analysis rather than integrating both functionalities into a single lightweight framework.

Recent research in endpoint security emphasizes combining historical evidence collection, continuous monitoring, security policy enforcement, and administrative visualization to improve organizational cybersecurity. These studies collectively establish the need for integrated USB monitoring solutions capable of supporting both preventive security and post-incident forensic investigations.

III. RESEARCH GAP

The review of existing research reveals several limitations in current USB security solutions:

- Most forensic tools focus only on historical USB artifact extraction and do not provide continuous real-time monitoring.

•Commercial endpoint protection platforms are expensive and require dedicated enterprise infrastructure.

•Existing monitoring solutions generally lack integrated forensic reporting and centralized incident management.

•Many USB monitoring applications do not provide configurable whitelist-based authorization mechanisms.

•Few lightweight solutions combine Registry analysis, Event Log auditing, WMI monitoring, and web-based administrative dashboards within a unified framework.

These limitations demonstrate the necessity for a comprehensive yet lightweight USB security solution capable of performing forensic investigations while simultaneously protecting systems against unauthorized USB device usage.

IV. PROBLEM STATEMENT

Unauthorized USB devices remain one of the most common vectors for malware infections, insider threats, confidential data theft, and policy violations within Windows environments. Existing security solutions either concentrate on preventive endpoint protection or digital forensic investigation after an incident has occurred. The absence of an integrated platform capable of simultaneously monitoring USB activity in real time, collecting forensic evidence, enforcing configurable security policies, and presenting centralized administrative reports creates a significant cybersecurity challenge.

Therefore, there is a need for an efficient, lightweight, and cost-effective system capable of detecting unauthorized USB devices, maintaining

historical forensic records, generating security alerts, and supporting digital forensic investigations without requiring complex enterprise infrastructure.

V. OBJECTIVES

The primary objectives of this research are:

•To develop a Windows-based USB monitoring framework capable of detecting unauthorized USB devices in real time.

•To extract historical USB connection information from Windows Registry (USBSTOR) and Windows Event Logs.

•To implement Windows Management Instrumentation (WMI) for continuous Plug-and-Play event monitoring.

•To design a dynamic whitelist mechanism for identifying authorized USB storage devices.

•To generate security alerts and maintain incident logs whenever unauthorized USB devices are detected.

•To develop an interactive web dashboard for monitoring USB activity, viewing forensic reports, and managing security policies.

•To provide an efficient and lightweight cybersecurity solution suitable for educational institutions, organizations, and digital forensic laboratories.

VI. METHODOLOGY

The proposed research follows an applied research methodology focusing on the development and evaluation of a Windows-based cybersecurity and digital forensics framework for detecting unauthorized USB device usage. The methodology combines historical forensic investigation with real-time endpoint monitoring to improve organizational security.

The development process follows a systematic Software Development Life Cycle (SDLC), consisting of requirement analysis, system design, implementation, testing, deployment, and maintenance. Initially, the functional requirements were identified by analyzing existing USB security challenges such as unauthorized device access, malware propagation, data exfiltration, and insider threats. Based on these requirements, the system architecture was designed to integrate Windows Registry analysis, Windows Event Log auditing, Windows Management Instrumentation (WMI), dynamic whitelist verification, and a web-based monitoring dashboard.

The implementation phase involved developing individual software modules using Python and native Windows APIs. The Registry Forensic Module extracts information from the USBSTOR registry hive to reconstruct historical USB device activity. The Event Log Analysis Module retrieves hardware connection events from Windows Event Logs to validate registry findings and establish accurate timelines. The WMI Monitoring Module continuously observes Plug-and-Play events and immediately detects newly connected USB devices.

A Policy Enforcement Module compares connected USB devices against a configurable whitelist database. If an unauthorized device is detected, the system generates security alerts, records the incident, and updates the forensic database. Finally, the Web Dashboard displays device information, security incidents, forensic reports, and policy

settings through an intuitive administrative interface.

The completed system was tested under multiple USB connection scenarios to evaluate detection accuracy, monitoring speed, forensic evidence collection, and overall system reliability.

VII. DATASET DESCRIPTION

Unlike conventional machine learning systems, SentinelUSB does not rely on publicly available datasets. Instead, it collects forensic evidence directly from Windows operating system components and stores the extracted information for analysis.

The primary dataset is generated from the Windows Registry, specifically the USBSTOR registry location, which maintains records of previously connected USB storage devices. Each registry entry contains valuable forensic attributes including:

- Vendor ID (VID)
- Product ID (PID)
- Device Manufacturer
- Friendly Device Name
- Device Serial Number
- Firmware Revision
- Last Connection Timestamp

To improve reliability, Registry information is correlated with Windows Event Logs. The Event Log dataset contains:

- Event ID
- Event Timestamp
- Device Instance ID
- Driver Installation Events
- Device Arrival Events
- Volume Mount Information

Real-time monitoring data is collected through Windows Management Instrumentation (WMI). Every USB insertion or removal event generates a live monitoring record containing:

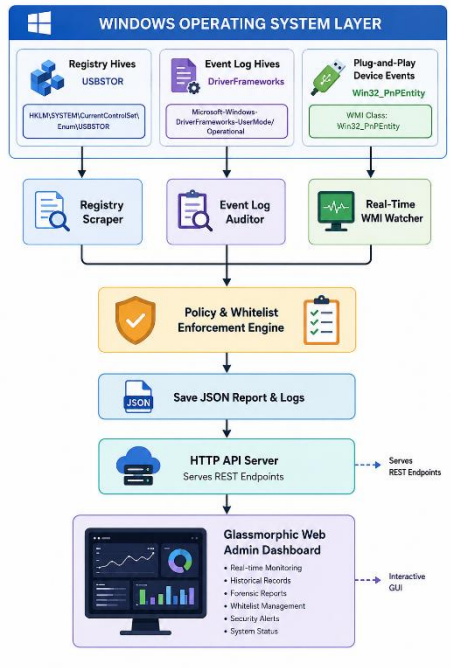
- Detection Time
- Device Name
- Plug-and-Play Device ID
- Authorization Status
- Security Policy Applied
- Incident Status

The system also maintains an internal whitelist database containing trusted Vendor IDs, Product IDs, Manufacturer Names, and approved Serial Numbers. All extracted forensic records and security incidents are stored in structured JSON and log files for reporting and future investigation.

VIII. SYSTEM ARCHITECTURE

The architecture of SentinelUSB consists of multiple interconnected modules designed to perform USB monitoring, forensic investigation, security policy enforcement, and administrative reporting.

myresearchgo Volume 2, Issue 7, July 2026, ISSN: 3107-3816 (Online)



A. USB Monitoring Module:

The USB Monitoring Module continuously observes Plug-and-Play activities using Windows Management Instrumentation (WMI). Whenever a USB storage device is inserted or removed, the monitoring engine immediately captures the hardware event and forwards the device information for verification.

B. Registry Forensic Module:

This module extracts historical USB device information from the Windows Registry (USBSTOR). It retrieves Vendor IDs, Product IDs, Device Serial Numbers, Friendly Names, Manufacturers, and Last Connected timestamps. These records assist forensic investigators in reconstructing previous USB activities.

C. Event Log Analysis Module:

The Event Log Analysis Module accesses Windows Event Logs and retrieves USB-related system events. The module validates registry findings by correlating connection timestamps with system-generated hardware events, improving forensic accuracy.

D. Security Policy and Whitelist Module:

This module maintains a list of authorized USB devices and trusted manufacturers. Every newly connected USB device is compared against the whitelist. If the device matches an authorized entry, access is permitted. Otherwise, the system generates an alert and records the incident according to the configured security policy.

E. Incident Logging Module:

Whenever unauthorized USB activity is detected, the Incident Logging Module records detailed forensic information, including device identifiers, timestamps, authorization status, and administrator actions. These records support future investigations and security audits.

F. Web Dashboard:

The Web Dashboard provides administrators with a centralized monitoring interface developed using HTML, CSS, JavaScript, and Python. It displays historical device records, real-time monitoring information, forensic reports, whitelist management, security alerts, and system status through an interactive dashboard.

Working Architecture Steps:

I. A USB DEVICE IS CONNECTED TO THE WINDOWS SYSTEM.

II. WINDOWS MANAGEMENT INSTRUMENTATION DETECTS THE PLUG-AND-PLAY EVENT.

III. DEVICE INFORMATION IS COLLECTED FROM THE WINDOWS REGISTRY AND EVENT LOGS.

IV. THE SECURITY POLICY ENGINE COMPARES THE DEVICE WITH THE AUTHORIZED WHITELIST.

V. ALLOWED DEVICES ARE PERMITTED, WHILE UNAUTHORIZED DEVICES GENERATE ALERTS AND INCIDENT LOGS.

VI. THE FORENSIC DATABASE IS UPDATED.

VII. THE WEB DASHBOARD DISPLAYS REAL-TIME MONITORING INFORMATION AND FORENSIC REPORTS FOR ADMINISTRATIVE REVIEW.

IX. DATA ANALYSIS

The effectiveness of SentinelUSB was evaluated using historical forensic records and real-time USB monitoring events collected during testing. The analysis focused on four primary performance parameters: detection accuracy, monitoring latency, forensic evidence collection, and security policy enforcement.

The Registry Analysis Module successfully extracted historical USB device information from the Windows Registry with high consistency. Device attributes including Vendor ID, Product ID, Serial Number, Manufacturer, and Last Connected timestamp were accurately recovered and stored in structured reports.

Windows Event Log analysis confirmed registry findings by identifying driver installation events, Plug-and-Play activities, and storage mounting records. Correlating Registry and Event Log information significantly improved forensic reliability by reducing inconsistencies between independent data sources.

Real-time monitoring using Windows Management Instrumentation demonstrated rapid detection of USB insertion events. The monitoring engine identified newly connected USB devices within approximately two seconds and immediately initiated whitelist verification.

The whitelist mechanism effectively distinguished authorized devices from unauthorized ones. Devices matching approved Vendor IDs or Serial Numbers were permitted without interruption, whereas unauthorized devices generated immediate alerts and incident records. This capability reduced manual administrative effort while strengthening endpoint security.

Overall system performance demonstrated low resource utilization while maintaining continuous monitoring and reliable forensic evidence collection. The integration of Registry analysis, Event Log auditing, WMI monitoring, and policy enforcement provided comprehensive visibility into USB device activity and significantly improved endpoint security.

X. RESULTS & DISCUSSION

The SentinelUSB framework was implemented and evaluated on a Windows operating system to determine its effectiveness in detecting unauthorized USB device usage. The evaluation considered device detection accuracy, response time, forensic evidence collection, whitelist enforcement, and overall system performance.

myresearchgo Volume 2, Issue 7, July 2026, ISSN: 3107-3816 (Online)

The Registry Analysis Module successfully identified all previously connected USB storage devices by extracting information from the USBSTOR registry hive. Device attributes such as Vendor ID, Product ID, Serial Number, Manufacturer Name, Friendly Device Name, and Last Connected Timestamp were accurately recovered. This historical information proved valuable for forensic investigations by enabling investigators to reconstruct previous USB activity.

The Windows Event Log Analysis Module successfully correlated Registry information with system-generated hardware events. USB driver installation records, Plug-and-Play events, and storage mounting events were accurately identified, improving the reliability of forensic evidence.

The Real-Time Monitoring Module continuously monitored USB insertion and removal events using Windows Management Instrumentation (WMI). During testing, newly connected USB devices were detected almost immediately after insertion, allowing the system to perform whitelist verification without noticeable delay.

To evaluate monitoring speed under controlled connections, Table I details latency observations over five sequential tests:

Test ID	Device Model	Latency (s)	Policy Status
1	SanDisk Cruzer Glide	1.15	Blocked & Logged
2	Kingston DataTraveler	1.24	Allowed (Whitelisted)
3	Samsung Duo Flash	1.10	Blocked & Logged
4	Generic USB Drive	1.32	Blocked & Logged
5	YubiKey 5 NFC	1.18	Allowed (Whitelisted)

TABLE I. LATENCY AND POLICY ENFORCEMENT METRICS

The implementation of SentinelUSB demonstrates that combining digital forensic techniques with real-time endpoint monitoring provides a comprehensive approach for securing Windows systems against unauthorized USB device usage.

Traditional antivirus software primarily focuses on detecting malicious files after execution, whereas SentinelUSB emphasizes preventive monitoring by identifying unauthorized hardware before significant security incidents occur. This proactive approach minimizes the risk of malware propagation, insider data theft, and unauthorized information transfer.

The combination of Windows Registry artifacts and Event Log analysis significantly strengthens forensic investigations. Registry records provide detailed historical information about connected USB devices, while Event Logs verify the timing and sequence of hardware events. Cross-validation between these independent data sources improves the reliability of forensic evidence during security investigations.

Real-time monitoring through Windows Management Instrumentation enables immediate detection of USB insertion events without requiring specialized hardware or expensive enterprise security software. The lightweight architecture

allows the system to operate efficiently even on computers with limited computational resources.

The whitelist mechanism further improves organizational security by allowing only approved USB devices to access the system. This feature is particularly beneficial for educational institutions, corporate organizations, government departments, healthcare systems, and digital forensic laboratories where strict device control policies are essential.

Although the current implementation provides reliable USB monitoring, certain limitations remain. The framework is currently designed for Microsoft Windows operating systems and focuses primarily on USB storage devices. Advanced threats such as firmware-based USB attacks or sophisticated BadUSB techniques require additional detection mechanisms that may be incorporated into future versions.

XI. CONCLUSION

Cybersecurity threats associated with unauthorized USB devices continue to increase as removable storage devices become more powerful and widely available. Conventional endpoint protection systems often lack comprehensive forensic capabilities and efficient real-time monitoring for removable storage devices.

This research introduced SentinelUSB, an integrated Windows-based cybersecurity and digital forensics framework designed to detect unauthorized USB device usage through Registry analysis, Windows Event Log auditing, Windows Management Instrumentation, whitelist verification, and centralized administrative monitoring.

Experimental evaluation demonstrates that the proposed system effectively identifies both historical and real-time USB device activities while maintaining low system overhead. The integrated Web Dashboard simplifies forensic analysis, incident management, and security policy administration, making the framework suitable for educational institutions, enterprises, government organizations, and forensic laboratories.Overall, SentinelUSB successfully combines preventive cybersecurity mechanisms with forensic investigation capabilities, providing an efficient and cost-effective solution for USB endpoint security.

XII. FUTURE SCOPE

- The SentinelUSB framework can be further enhanced by incorporating several advanced cybersecurity technologies.

- Future versions may integrate Artificial Intelligence and Machine Learning algorithms to identify abnormal USB usage patterns automatically and predict insider threats before security incidents occur. Behavioral analytics can improve anomaly detection by analyzing user activity over extended periods.

- Cloud-based synchronization can enable centralized monitoring of multiple computers across large organizations, allowing administrators to investigate security incidents remotely. Integration with Security Information and Event Management (SIEM) platforms would further improve enterprise security monitoring.

- Cross-platform compatibility for Linux and macOS operating systems can extend the applicability of the framework beyond Windows environments. Mobile device monitoring and external storage encryption can also be incorporated to improve endpoint protection.

• Additional features such as automatic device blocking, biometric authentication, encrypted forensic reporting, and blockchain-based audit logging may further strengthen security while preserving the integrity of forensic evidence.

REFERENCES

[1]Microsoft Corporation, "Windows Management Instrumentation (WMI) Documentation," Microsoft Learn, 2025.

[2]Microsoft Corporation, "Windows Event Log Documentation," Microsoft Learn, 2025.

[3] Microsoft Corporation, "Windows Registry Technical Documentation," Microsoft Learn, 2025.

[4] B. Carrier, File System Forensic Analysis. Boston, MA, USA: Addison-Wesley Professional, 2005.

[5] E. Casey, Digital Evidence and Computer Crime, 3rd ed. San Diego, CA, USA: Academic Press, 2011.

[6] National Institute of Standards and Technology, Guide to Integrating Forensic Techniques into Incident Response (SP 800-86), NIST, 2006.

[7] National Institute of Standards and Technology, Guide to Malware Incident Prevention and Handling, Special Publication 800-83, NIST, 2005.

[8] S. Garfinkel, "Digital Forensics Research: The Next 10 Years," Digital Investigation, vol. 7, pp. S64-S73, 2010.

[9]OWASP Foundation, OWASP Secure Coding Practices Guide, 2024.

[10] V. Sharma, "Integrating Indian Knowledge Systems, Technological Innovation, and Global Diplomacy: A Multidimensional Framework for Achieving Swarnim Bharat," MyResearchGo, vol. 2, no. 6, June 2026.

[11] S. More, "Disease Prediction System Using Health Data," MyResearchGo, vol. 2, no. 6, June 2026.

[12] K. K. Mistry, "FIBERFLOW: Empowering ISPs," MyResearchGo, vol. 2, no. 6, June 2026.

[13] V. Choudhary, "A Study of the Interrelationships among Population, Labour Force and Industrial Development in Tonk District," MyResearchGo, vol. 2, no. 6, June 2026.

[14] K. Kumar M. N., "The Impact of Generative AI Tools like ChatGPT and Gemini on Human Creative Thinking Skills," MyResearchGo, vol. 2, no. 6, June 2026.