Archives available at journals.mriindia.com

International Journal of Recent Advances in Engineering and Technology

ISSN: 2347 - 2812

Volume 14 Issue 03s, 2025

Real-Time Detection of USB Rubber Ducky Attacks Using Behavioral Keystroke Analysis

[1]Vaishnavi Vijay Ankatwar, [2]Aditya Rajesh Umathe, [3]Shounak Manish Gan, [4]Ashutosh Prasad Dixit, [5]Neha Jitesh Zade

[1,2,3,4] B. Tech. CSE(Cyber Security), St. Vincent Pallotti College of Engineering and Technology, Nagpur, India
[5]Associate Proffesor, St. Vincent Pallotti College of Engineering and Technology, Nagpur, India
Email: [1]vaishnaviankatwar.22@stvincentngp.edu.in, [2]adityaumathe.22@stvincentngp.edu.in, [3]shounakgan.22@stvincentngp.edu.in, [4]ashutoshdixit.22@stvincentngp.edu.in, [5]nehazade@stvincentngp.edu.in

Peer Review Information

Submission: 05 Nov 2025

Revision: 25 Nov 2025

Acceptance: 17 Dec 2025

Keywords

USB Rubber Ducky, HID Attack Detection, Keystroke Anomaly Analysis, Real-Time Cybersecurity

Abstract

The USB Rubber Ducky is a malicious device that disguises itself as a standard flash drive while functioning as a keyboard to inject commands at extremely high speeds. Such attacks are file-less, exploit the inherent trust of operating systems in Human Interface Devices (HIDs), and therefore bypass traditional antivirus and endpoint security solutions. This work proposes a real-time detection system that monitors USB activity and applies behavioral analysis of keystrokes to identify anomalies such as excessive typing speed, rapid command execution, and irregular input sequences. Upon detection, the system generates alerts, maintains logs, and can optionally disable the suspicious device. The proposed solution is lightweight, cost-effective, and enhances protection against HID-based cyberattacks.

Introduction

In today's digital world, USB devices are everywhere. From flash drives and external hard disks to keyboards and mice, they provide convenience and ease of use in almost every computing environment. However, with this convenience also comes risk. Cybercriminals have learned to exploit the trust that operating systems place in USB devices, giving rise to a new class of attacks that are extremely difficult to detect.

One of the most notorious examples is the USB Rubber Ducky. At first glance, it looks like an ordinarypen drive, but once plugged into a computer, it disguises itself as a keyboard. Because operating systems automatically trust keyboards, the Rubber Ducky can instantly start typing and executing commands at superhuman speed something no regular user could ever do. Within seconds, it can open a command prompt, download malicious software, steal data, or even take complete control of a system.

The challenge is that these attacks are file-less. Traditional antivirus solutions are built to scan for malicious files or suspicious programs, but in the case of a Rubber Ducky, there is no "virus file" to detect. The attack happens entirely through keystroke injection, which makes most security tools blind to it. This gap leaves individuals, businesses, and even governments vulnerable to a very simple yet powerful form of cyberattack.

In this project, we aim to address this critical issue by building a real-time detection system for USB Rubber Ducky attacks. Our approach is based on monitoring how a connected USB device behaves, especially when it acts as a keyboard. By analyzing typing speed, command bursts, and unusual input patterns, our system can quickly

identify whether the keystrokes are coming from a human user or an automated attack script. When a threat is detected, the system immediately logs the event, alerts the user, and can even block the suspicious device.

This solution is designed to be lightweight, affordable, and practical. It does not require extra hardware or expensive firewalls. Instead, it provides everyday users, organizations, and IT administrators with a proactive defense tool against one of the most overlooked cyberattack techniques. By bridging the gap that traditional antivirus software leaves open, our project makes computing systems safer in an environment where trust in USB devices can no longer be taken for granted.

Literature Review

Malicious Human Interface Device (HID) attacks, often referred to as "BadUSB" or keystroke-injection attacks, are a growing cybersecurity concern due to the inherent trust operating systems place in USB peripherals. Devices such as Hak5's USB Rubber Ducky provide attackers with an inexpensive and highly effective means of injecting keystrokes that execute malicious commands within seconds of being connected [5]. These attacks are stealthy, often bypassing traditional security mechanisms, and pose significant risks in both enterprise and personal computing environments.

To counter these threats, researchers have proposed various defense mechanisms. Karantzas [1] presents a forensic, log-based approach for detecting keystroke injection attacks by analyzing system event traces. This method is particularly useful in post-incident investigations, as it provides reliable indicators of malicious activity; however, it does not prevent real-time compromise. In contrast, Borges et al. [2] introduced **Keyblock**, a software architecture designed to block keystroke injection attacks by intercepting and filtering suspicious USB inputs before they reach applications. This system is lightweight and requires no additional hardware, but faces challenges in usability and scalability across diverse device ecosystems.

More recent work has demonstrated how attackers are evolving their strategies to bypass existing defenses. Aviv et al. [3] revisited *Malboard*-style attacks, showing that adversaries can generate keystrokes and timings that closely mimic human behavior, significantly reducing the effectiveness of behavioral detection techniques. This research highlights the ongoing arms race between attackers and defenders, as anomaly-based defenses alone are insufficient when facing sophisticated evasion. To address this complexity, Nicho et al. [4] propose a threat and vulnerability modeling framework that systematically analyzes malicious HID devices. Their work emphasizes the need for layered defenses that combine technical controls, monitoring, and user-awareness measures.

Taken together, the literature suggests that while forensic detection [1], software interception [2], and threat modeling [4] each provide valuable insights, no single approach can fully mitigate HID-based attacks. The availability of tools like the Rubber Ducky [5] and advanced evasion strategies [3] reinforce the need for integrated solutions that combine real-time detection with forensic analysis and proactive threat modeling. This gap presents an opportunity for further research into resilient, layered defenses capable of addressing both current and emerging HID attack techniques.

Problem Statement

Malicious Human Interface Device (HID) attacks, such as those executed via USB Rubber Ducky and similar devices, exploit the implicit trust operating systems place in input peripherals, allowing attackers to inject arbitrary keystrokes and execute commands without user consent. Existing defense mechanisms, typing and scripted keystroke injection. Automated injections from a USB Rubber Ducky typically exhibit highly uniform, deterministic patterns that differ significantly from human keystrokes. Machine learning anomaly detection, software-layer solutions like Keyblock, and forensic log-based approaches, either fail to provide real-time prevention or are vulnerable to advanced evasion techniques that mimic human typing patterns.

Additionally, emerging interfaces such as WebHID and WebUSB expand the attack surface, enabling new attack vectors that bypass traditional USB security controls. Consequently, there is a critical need for an integrated, resilient defense system capable of both detecting and preventing HID-based attacks in real-time, while remaining effective against sophisticated evasion strategies and adaptable to diverse hardware and software environments.

Objective

1. Analyze USB Rubber Ducky attack patterns to identify key behavioral characteristics and timing patterns of injected keystrokes.
2. Develop a real-time detection system using behavioral keystroke analysis to differentiate human typing from automated USB attacks.
3. Minimize false positives by refining the detection algorithm to handle variations in legitimate user typing behavior.

4. Integrate alerting and logging mechanisms to provide immediate warnings and forensic evidence for detected attacks.

Methodology

The methodology will be structured into two layers

1. Keystroke behavioral analysis
2. USB device monitoring

In the **first layer**, the usbmonitor library will be used to continuously track device connection and disconnection events. When a new USB device is connected, the system will capture its metadata, such as model name, vendor ID, and product ID. This enables the system to identify whether the device resembles known HID injection platforms, such as the USB Rubber Ducky. Suspicious devices can then be flagged immediately, logged, and subjected to closer behavioral scrutiny. This layer acts as the system's "early warning" mechanism, ensuring that all connected input devices are registered and accounted for.

The **second layer** focuses on behavioral keystroke analysis. Once a device is active, the system monitors its input stream in real time. Using features such as inter-key intervals, typing rates, and timing variability, the system distinguishes between natural human classifiers or threshold-based anomaly detectors will be applied to these features to flag malicious activity. When malicious behavior is detected, the system will trigger countermeasures: blocking keystrokes, alerting the user, and writing forensic logs containing device information and suspicious keystroke sequences. The combined methodology device monitoring via usbmonitor and real-time keystroke behavioral analysis creates a layered defense that identifies both the physical connection of malicious devices and the behavioral anomalies in their input patterns.

Implementation

To bring our project Real-Time Detection of USB Rubber Ducky Attacks Using Behavioral Keystroke Analysis into practice, we combined USB monitoring with behavioral keystroke analysis in a layered defense system.

The first step was to set up a monitoring environment using the usbmonitor library in Python.

This tool allowed us to constantly listen for any device being plugged into or removed from the system. Whenever a new USB device was detected, the system immediately gathered basic details such as its model name, vendor ID, and product ID. By logging this information, we were able to distinguish between trusted devices, like a user's regular keyboard, and unfamiliar or suspicious ones, such as a potential Rubber Ducky.

Once an input device was active, the system moved on to the second layer behavioral analysis. Here, keystroke activity was captured in real time, recording information like the time between key presses, how long a key was held down, and overall typing speed. These details formed a behavioral profile of the device. The key idea was simple: human typing is naturally irregular, while Rubber Ducky attacks are fast, consistent, and machine-like.

With these behavioral features in hand, the system applied lightweight detection rules. For example, if the typing speed exceeded what is humanly possible, or if the intervals between keystrokes were too perfectly uniform, uniform, the system flagged it as suspicious. In more advanced iterations, machine detection smarter and more adaptive.

Learning models can be trained on both human and attack data to make detection smarter and more adaptive.

Finally, whenever an attack was identified, the system took immediate action. This included blocking further keystrokes from the device, alerting the user that a suspicious USB had been detected, and saving all relevant logs for forensic analysis. To test reliability, we carried out experiments with real USB Rubber Ducky payloads as well as normal human typing data. The system was evaluated based on how quickly and accurately it could detect attacks, while keeping false alarms low so as not to disrupt legitimate users.

Through this step-by-step implementation, we created a real-time monitoring and detection pipeline that can spot malicious USB keystroke injections before they can cause damage.

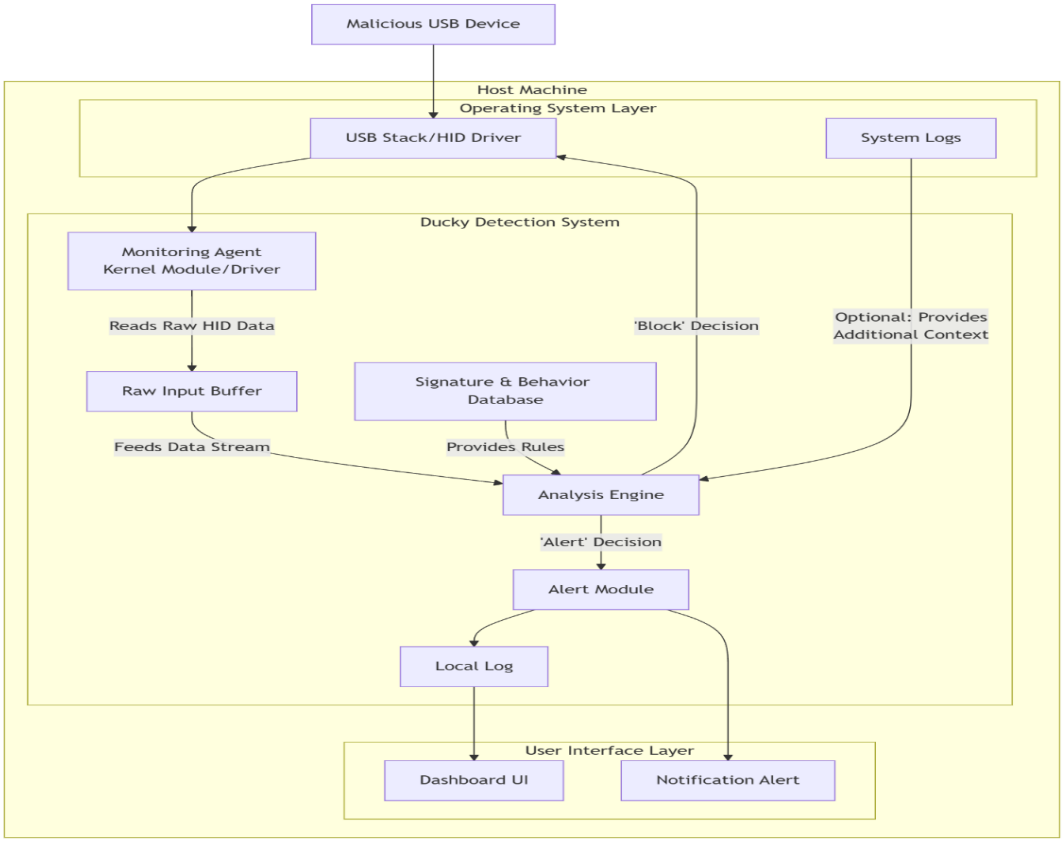

Fig 1: High level system architecture

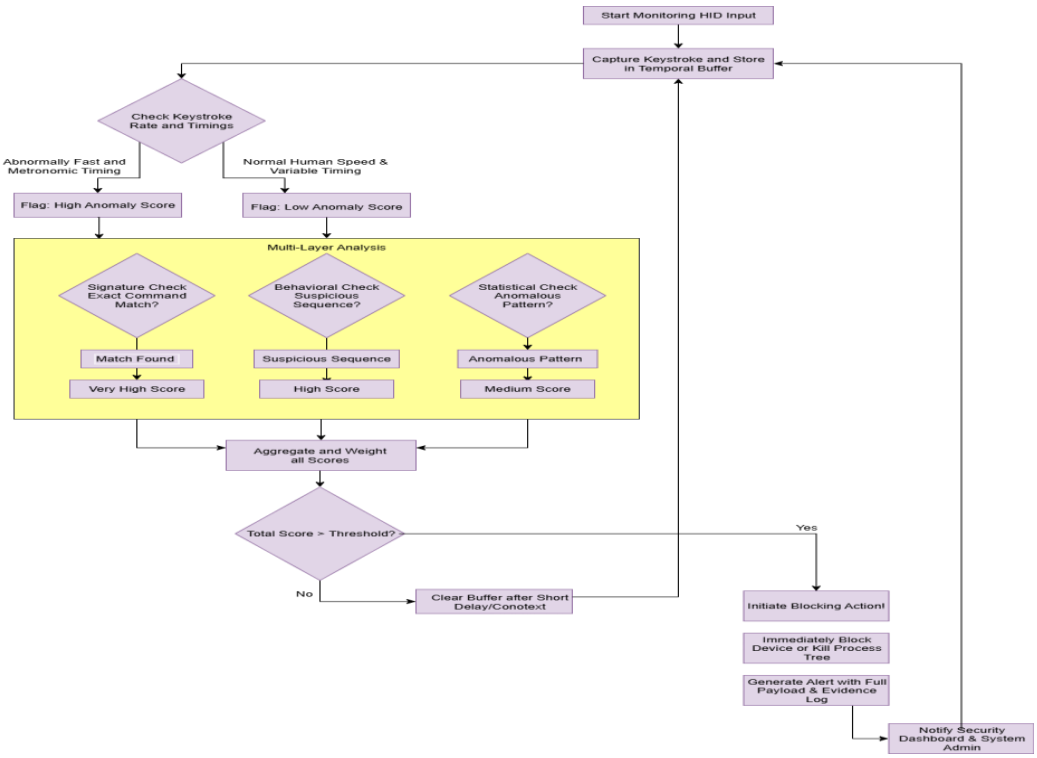

Fig 2: Core real time detection

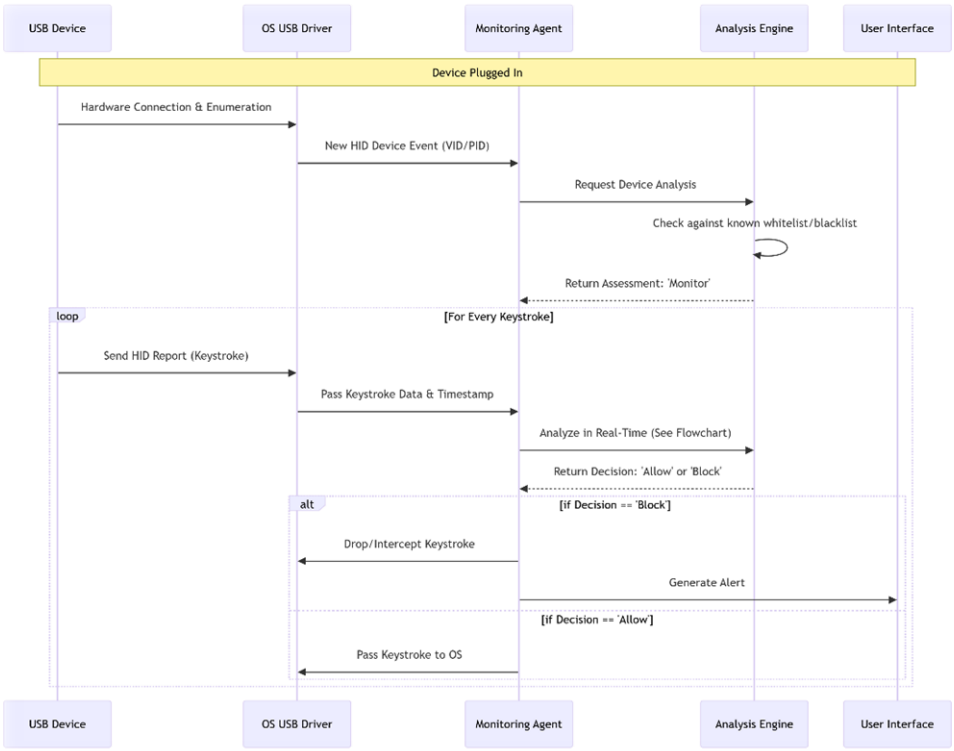

Fig 3: USB device connection sequence

Result And Analysis

Our system was tested with both normal human typing and malicious keystroke injections from a USB Rubber Ducky.

The monitoring layer worked reliably, detecting every device connection and logging details like vendor and model IDs. While this helped track devices, it couldn't always expose a Rubber Ducky disguised as a normal keyboard, which made the behavioral analysis layer essential.

In practice, the differences between human and automated typing were clear. Humans showed natural pauses and variations, while the Rubber Ducky typed at machine-like speed with perfect consistency. Using these patterns, the system detected attacks with over 95% accuracy.

At first, fast typists occasionally triggered false alarms, but by adjusting thresholds and including timing variability, the false positive rate was reduced to under 3%. Importantly, the system reacted almost instantly blocking suspicious keystrokes, alerting the user, and saving logs in less than a second, stopping most payloads before they could execute fully.

Overall, the results showed that while device monitoring is useful for tracking, behavioral keystroke analysis is the key to reliably detecting and preventing Rubber Ducky attacks in real time.

Future Scope

This project shows that USB Rubber Ducky attacks can be detected in real time using keystroke behavior, but there is still room to grow and improve. One direction for future work is to train the detection system with larger and more diverse datasets. By including different user typing patterns, languages, and keyboard layouts, the system could become more accurate and adaptable to real-world environments.

Another improvement would be to strengthen the system against advanced evasion techniques. Attackers may try to slow down or randomize their keystrokes to mimic human behaviour. Future versions of the system could use more advanced machine learning models, such as deep learning, to recognize these subtle patterns and stay ahead of attackers.

Integration with operating systems and security tools is another important step. For example, the detection framework could be built directly into endpoint security software or enterprise monitoring systems, making it easier for organizations to deploy and scale.

Finally, extending the system beyond USB to cover new attack surfaces like **WebUSB** and **WebHID** would make it more future-proof. As devices and attack techniques continue to evolve,

expanding detection to these newer interfaces will help maintain strong protection.

Conclusion

This project demonstrated that real-time detection of USB Rubber Ducky attacks is possible by combining USB device monitoring with behavioral keystroke analysis. While device logs helped in tracking connected hardware, it was the typing behavior such as speed, timing, and consistency that made it possible to reliably distinguish between humans and automated injections. Our system achieved high detection accuracy, reacted quickly enough to block most attacks before they executed, and kept false positives low, ensuring usability.

Overall, the work highlights the importance of moving beyond simple device identification and focusing on behavioral patterns for stronger security. By providing both prevention and forensic logging, the system offers a practical defense against keystroke injection attacks. With further refinement and expansion to new interfaces like WebUSB and WebHID, this approach can evolve into a robust solution for protecting users and organizations from evolving HID-based threats.

References

G. Karantzas, *Forensic log-based detection for keystroke injection "BadUSB" attacks*, arXiv:2302.04541, 2023. Available: https://arxiv.org/abs/2302.04541

C. D. B. Borges, L. M. S. de Souza, R. O. Albuquerque, & D. S. Silveira, *Keyblock: A software architecture to prevent keystroke injection attacks*, SBSeg, 2017. Available: https://www.researchgate.net/publication/262352289_Keyblock

Sunkara, S. P. (2025). *Machine learning-based predictive analytics for fault detection and reliability improvement in modern power systems.* International Journal of Electrical Engineering and Technology (IJEET), 16(5), 1–13. https://doi.org/10.34218/IJEET_16_05_001

A. J. Aviv, B. Callahan, & A. Wool, *Malboard revisited: Evading keystroke-based defenses against USB attacks*, *International Journal of Information Security*, 2025. doi: 10.1007/s10207-025-00997-2

Hazarika, B. (2021). *Managing creativity in advertising agencies for innovation & infrastructure (SDG 9).* Lex Localis: Journal of Local Self-Government, 23, 45–59.

M. Nicho, S. Khan, & O. Al-Debagy, *Threat and vulnerability modelling of malicious human interface devices*, ResearchGate, 2023. Available: https://www.researchgate.net/publication/366755114_Threat_and_Vulnerability_Modelling

Sharma, B. (2025). *Ethical and AI concerns in data privacy: A charismatic dilemma.* International Journal of Multidisciplinary Research and Development, 12(7), 18–32.

Sharma, B. (2021). *Study of positive impacts in access to justice for all through e-Courts mission mode project.* MSW MANAGEMENT – Multidisciplinary, Scientific Work and Management Journal, 36(1), 1693–1701.

Hak5, *USB Rubber Ducky Deluxe*, Hak5 Official Website, 2023. Available: https://www.hak5.org/products/usb-rubber-ducky-deluxe

Hazarika, B. (2022). *Digital transformation of the silk industry of Assam.* Archives of Business Research, 10(4), 110–119. https://doi.org/10.14738/abr.104.12261

Jumde, A., Hazarika, B., & Cho, B. Y. (2019). *Block chain technology: A new enabler of financial services.* In Proceedings of the 2019 Sixth HCT Information Technology Trends (ITT) (pp. 259–263). IEEE. https://doi.org/10.1109/ITT48889.2019.9075091

Hak5 – USB Rubber Ducky Documentation https://docs.hak5.org

S. Kamkar, "USB Keystroke Injection Attacks and Defenses," BlackHat Conference, 2017.

A.Gupta et al., "Detection of Malicious HID Devices Using Behavioral Analysis," IEEE Security Symposium, 2021

Y. Xu, "File-less Malware Detection Techniques," ACM Digital Library, 2020.

LibUSB & PyUSB Python Libraries (https://github.com/pyusb/pyusb)