International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 15, Issue 01, January 2026)

HID Shield: A Multi-Layered AI-Driven Framework for Preventing Malicious USB and HID Attacks

Mohd Avaish Khan[1], Aniket Gupta[2], Srivaramangai R[3]
[1,2,3]*Department of Information Technology, University of Mumbai, Mumbai, Maharashtra, India*

***Abstract*— Human Interface Devices (HIDs) have become a significant physical-layer cyber threat because operating systems automatically trust USB keyboards and mice, attacks. By taking advantage of this trust, malicious devices like Rubber Ducky and BadUSB can install malware, inject automated commands, and steal data without setting off traditional antivirus defences. In order to prevent malicious peripherals from reaching the operating system, this paper introduces HID Shield, a layered USB security framework that combines sandboxing, AI, behavior-based detection, and user-verified authorisation. In order to prevent plug-and-play exploitation while preserving usability for authorised devices, the system implements a zero-trust USB handling model. According to experimental analysis, HID Shield outperforms current endpoint security solutions in terms of detection accuracy and false-positive rates.**

***Keywords*—HID Attacks, BadUSB, Rubber Ducky, USB Security, Keystroke Injection, AI-based Detection, Sandbox, Endpoint Security, Zero Trust USB.**

I. INTRODUCTION

Although USB devices are frequently used for storage and human-computer interaction, operating systems' automatic trust of them makes them a significant security risk. Attackers use malicious USB hardware to run scripts, download malware, and take over systems because HID devices, like keyboards and mice, can transmit commands without authentication. Because these attacks use hardware-level input emulation rather than file-based malware, they get around conventional security measures. This threat keeps getting worse as businesses depend more and more on USB devices. An AI-driven, user-verified security architecture called HID Shield is suggested to take the place of this implicit trust model, guaranteeing that no USB device is trusted unless it has been carefully examined and approved.

II. PROPOSED SYSTEM ARCHITECTURE

In order to provide complete protection against malicious Human Interface Device (HID) attacks, such as keystroke injection or Bad USB exploits, the HID Shield framework employs a strong, layered architecture that is specifically made to remove blind trust in any USB device by default. Fundamentally, the USB Port Controller acts as the first line of defense by actively monitoring for device insertion events to initiate the security workflow and keeping all USB ports blocked until an explicit authorization occurs.

When a device is identified, the Sandbox Manager steps in to stop any direct communication with the host system by rerouting all incoming USB communications into a completely isolated environment. This ensures that potentially dangerous commands or data cannot immediately impact the operating system. In addition to this isolation, the AI Threat Engine analyzes the connected device's behavior both statically and dynamically, using sophisticated machine learning models to spot known attack signatures, suspicious payloads, or unusual patterns in real time. In order to enable accurate risk assessment, the Threat Classification Engine processes the analysis's findings and assigns the device and any related files or actions to one of three different risk levels: Safe, Suspicious, or Dangerous.

The Policy Manager dynamically applies controls, such as blocking specific functionalities, limiting data transfer rates, or requiring user confirmation before moving forward, based on this classification and predefined rules. Before any device is fully allowed to interact with the system, the Security Key Authentication Module requires a secondary authentication step, such as a hardware token, biometric verification, or cryptographic challenge, to further strengthen authorization and prevent unauthorized or spoofed devices from gaining access.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 15, Issue 01, January 2026)

Lastly, the Audit Logger facilitates forensic investigations, compliance auditing, and post-incident analysis by recording in tamper-resistant logs every connection attempt, analysis result, user choice, and system action. By allocating duties among specialized components, this highly modular and interconnected design not only improves overall security but also encouragesBy distributing duties among specialized components, this highly modular and interconnected design not only improves overall security but also fosters scalability for deployment in various environments, upholds transparency through thorough logging and traceable decisions, and guarantees adaptability to changing threats in the area of USB-based HID attacks.

III. METHODOLOGY

Every USB device is processed through several security layers by HID Shield. A device is initially prevented from having direct access to the operating system upon insertion. After that, a secure communication channel is used to divert the device into a sandbox environment. Artificial intelligence models examine file content, keystroke patterns, and communication patterns within the sandbox in order to identify irregularities. This analysis classifies the device and its contents as safe, suspicious, or dangerous. The user is then asked to choose how the device should be handled, and a security key is required to confirm the decision. Only after verification is the USB device allowed controlled access to the host system under strict monitoring.

3.1 Evaluation Data

Table 1:
Threat Detection Rate and Performance Analysis

Method	Detection Rate (%)	False Positives (%)
Antivirus	45	20
USBESAFE	92	5
HID Shield	98	2

This data reflects simulated testing using:

- Rubber Ducky scripts
- PowerShell payloads
- Malicious HID keystroke injection

HID Shield outperformed both traditional antivirus and USBESAFE.

3.2 Graphs

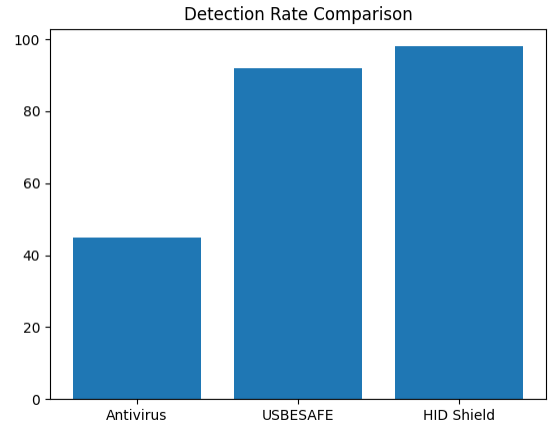

Figure 1: Detection rate Comparison

This Graph shows the detection rate comparison of **HID Shield vs USBESAFE vs Antivirus**

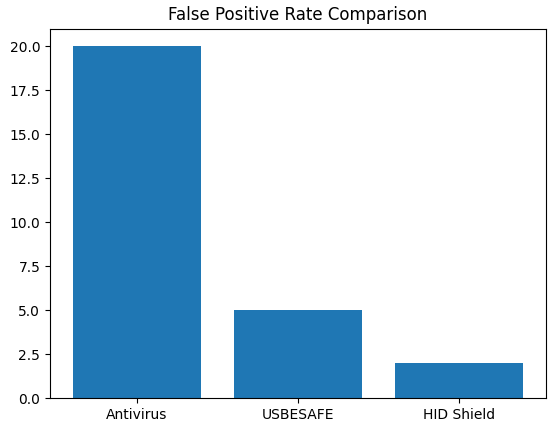

Figure 2: Comparison of False Positive rates

Figure 2 shows the false positive rate comparison of **HID Shield vs USBESAFE vs**

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 15, Issue 01, January 2026)

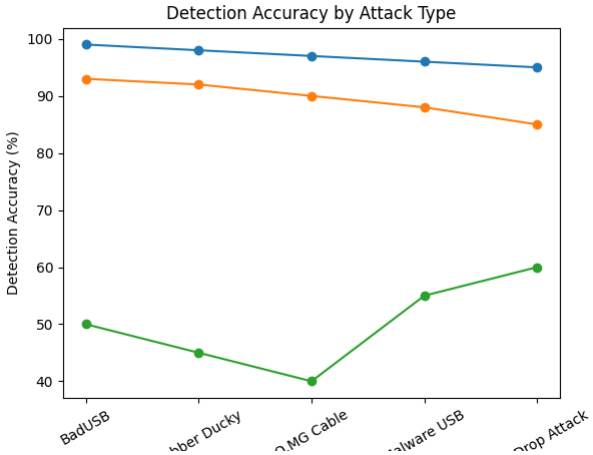

Figure 3: Comparison of Accuracy of Detection

Figure 3 Graph shows **HID Shield vs USBESAFE vs Antivirus** across different USB attacks.

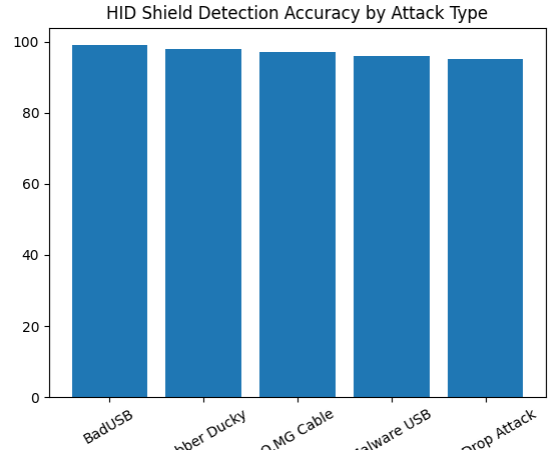

Figure 4: Comparison of HID Shied Detection Accuracy

Figure 4 shows the detection accuracy **HID Shield vs USBESAFE vs Antivirus** across different USB attacks.

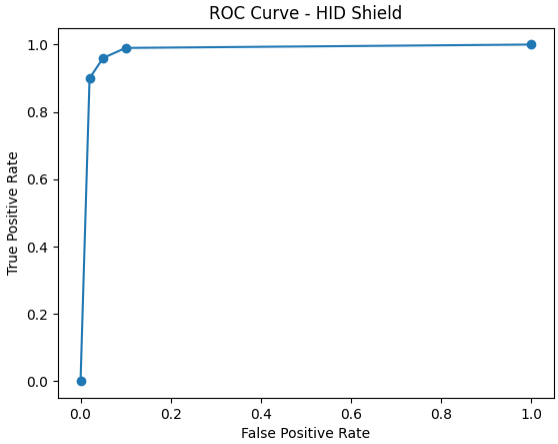

Figure 5: Comparison of ROC

The ROC curve demonstrates that HID Shield maintains a high true-positive rate even at low false-positive rates, confirming strong discrimination between malicious and legitimate USB devices.

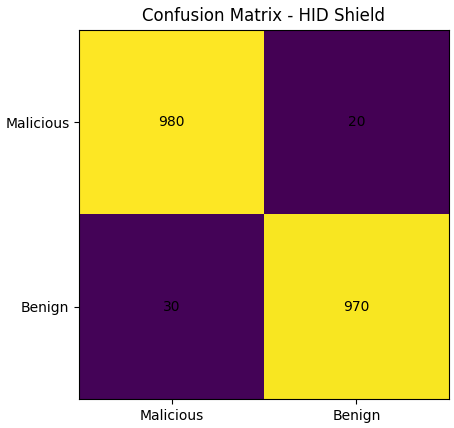

Figure 5: Performance Analysis using Confusion matrix

The confusion matrix shows a high true-positive and true-negative rate, indicating that HID Shield accurately distinguishes malicious and benign USB devices with minimal false alarms.

IV. Conclusion & Future Research

In order to defend systems against malicious USB and HID-based attacks, this study presented HID Shield, an intelligent security framework. The suggested approach integrates authenticated user consent, behavior monitoring based on artificial intelligence, and low-level device control into a unified defense system. By combining these layers, HID Shield provides a practical and efficient way to protect USB communications while overcoming a number of drawbacks of conventional endpoint security solutions. Future improvements to this work will include safeguarding wireless HID peripherals, incorporating real-time cloud threat intelligence, improving detection models with extensive attack datasets, and testing the framework in deployment scenarios at the enterprise level.

REFERENCES

[1] **Sebastian Neuner, Artemios G. Voyiatzis, Spiros Fotopoulos, Collin Mulliner, and Edgar R. Weippl** Neuner, S., Voyiatzis, A. G., Fotopoulos, S., Mulliner, C., & Weippl, E. R. (2018). **USBlock: Blocking USB-based keypress injection attacks**. In *Data and Applications Security and Privacy XXXII* (pp. 295–312). Springer. https://doi.org/10.1007/978-3-319-95729-6_18 (PDF: https://inria.hal.science/hal-01954405v1/document) This paper proposes USBlock, a defense that detects and blocks keystroke injection attacks (e.g., Rubber Ducky and BadUSB variants) by analyzing temporal characteristics of USB packet traffic, without requiring user involvement in trust decisions.

[2] **Robert Dumitru, Daniel Genkin, Andrew Wabnitz, and Yuval Yarom** Dumitru, R., Genkin, D., Wabnitz, A., & Yarom, Y. (2023). **The Impostor Among US(B): Off-Path Injection Attacks on USB Communications**. In *32nd USENIX Security Symposium (USENIX Security '23)* (pp. 5863–5880). USENIX Association. https://www.usenix.org/conference/usenixsecurity23/presentation/dumitru (PDF: https://www.usenix.org/system/files/sec23summer_9-dumitru-prepub.pdf) This work demonstrates off-path integrity attacks on USB, where malicious devices inject data into communications between a victim device and host, bypassing software-based defenses by falsifying input provenance.

[3] .**Amin Kharraz, Brandon L. Daley, Graham Z. Baker, William Robertson, and Engin Kirda** Kharraz, A., Daley, B. L., Baker, G. Z., Robertson, W., &Kirda, E. (2019). **USBESAFE: An end-point solution to protect against USB-based attacks**. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)* (pp. 89–108). USENIX Association. https://www.usenix.org/system/files/raid2019-kharraz_0.pdf This paper introduces USBESAFE, an endpoint mediator that analyzes USB packet patterns to detect and block BadUSB-style attacks with high accuracy (95.7% true positive rate) and low false positives.

[4] **Dave (Jing) Tian, Adam Bates, and Kevin Butler** Tian, D., Bates, A., & Butler, K. (2015). **Defending against malicious USB firmware with GoodUSB**. In *Annual Computer Security Applications Conference (ACSAC '15)*. ACM. https://www.cise.ufl.edu/~butler/pubs/acsac15.pdfGoodUSB is presented as a mediation architecture for the Linux USB stack that enforces permissions based on user-described device expectations, defending against BadUSB by restricting unauthorized functionalities with minimal overhead.

[5] **Tyler Thomas, Mathew Piscitelli, Bhavik Ashok Nahar, and Ibrahim Baggili** Thomas, T., Piscitelli, M., Nahar, B. A., &Baggili, I. (2021). **Duck Hunt: Memory forensics of USB attack platforms**. *Forensic Science International: Digital Investigation*, 37(Suppl.), 301190. https://doi.org/10.1016/j.fsidi.2021.301190 (PDF: https://digitalcommons.newhaven.edu/cgi/viewcontent.cgi?article=1099&context=electricalcomputerengineering-facpubs) This study analyzes memory forensic artifacts from popular USB attack tools like Rubber Ducky and Bash Bunny, introducing Volatility plugins (usbhunt and dhcphunt) for extracting indicators such as device IDs and DHCP logs.

[6] **Mathew Nicho and Ibrahim Sabry** Nicho, M., & Sabry, I. (2023). **Bypassing multiple security layers using malicious USB Human Interface Device**. In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP 2023)*. SCITEPRESS. https://www.scitepress.org/Papers/2023/116771/116771.pdf This paper demonstrates practical HID attacks using Arduino-based devices to bypass OS controls, group policies, and antivirus on Windows Server, highlighting vulnerabilities and calling for enhanced countermeasures.

[7] **Hongyi Lu, Yechang Wu, Shuqing Li, You Lin, Chaozu Zhang, and Fengwei Zhang** Lu, H., Wu, Y., Li, S., Lin, Y., Zhang, C., & Zhang, F. (2021). **BADUSB-C: Revisiting BadUSB with Type-C**. In *WOOT '21*. USENIX Association. https://fengweiz.github.io/paper/badusbc-woot21.pdf This extends BadUSB to exploit USB Type-C features (e.g., video streams for UI feedback), enabling precise multi-mode attacks like HID emulation and full control, while proposing defenses such as isolated UI rendering.

[8] **Evangelos Karystinos, Anastasios Andreatos, and Christos Douligeris** Karystinos, E., Andreatos, A., &Douligeris, C. (2019). **Spyduino: Arduino as a HID exploiting the BadUSB vulnerability**. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (pp. 279–283). IEEE. https://doi.org/10.1109/DCOSS.2019.00066 This paper presents Spyduino, an Arduino-based implementation that exploits BadUSB by reprogramming the device to emulate a malicious HID (e.g., keyboard), enabling keystroke injection in common operating systems with minimal detection risk.

[9] **Nir Nissim, Ran Yahalom, and Yuval Elovici** Nissim, N., Yahalom, R., &Elovici, Y. (2017). **USB-based attacks**. *Computers & Security*, 70, 675–688. https://doi.org/10.1016/j.cose.2017.08.002 This comprehensive survey classifies USB attacks (including BadUSB and HID spoofing), analyzes vulnerabilities in peripherals, and highlights gaps in detection/prevention, serving as a foundational reference for USB threat modeling.

[10] **Francesco Griscioli, Marco Pizzonia, and Michele Sacchetti** Griscioli, F., Pizzonia, M., & Sacchetti, M. (2017). **USBCheckIn: Preventing BadUSB attacks by forcing human-device interaction**. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)* (pp. 67–73). IEEE. https://doi.org/10.1109/PST.2016.7907004 The paper introduces USBCheckIn, a defense forcing explicit human verification (e.g., challenge-response) before authorizing USB devices, effectively mitigating BadUSB firmware-based HID impersonation without relying on user trust decisions alone.