

HID PAYLOAD INJECTING TOOL

1st KOMAL MEHTA

Assistant Professor
Department of AIT CSE
Chandigarh University,
Mohali, Punjab, India
komal.e15888@cumail.in

2nd ROBERT

Department Of AIT CSE
Chandigarh University,
Mohali, Punjab, India
UID- 22BIS50001
22bis50001@cuchd.in

3rd SHUBHAM PATEL

Department of AIT CSE
Chandigarh University,
Mohali, Punjab, India
UID- 22BIS50002
22bis50002@cuchd.in

4th LALIT

Department of AIT CSE
Chandigarh University,
Mohali, Punjab, India
UID- 22BIS50004
22bis50004@cuchd.in

5th SHASHWAT

Department of AIT CSE
Chandigarh University,
Mohali, Punjab, India
UID- 22BIS50005
22bis50005@cuchd.in

6th HRITIK JAIN

Department of AIT CSE
Chandigarh University,
Mohali, Punjab, India
UID- 23BIS80004
23bis80004@cuchd.in

Abstract— Human Interface Device (HID) Payload Injecting tool is a cybersecurity instrument that leverages the inherent trust of USB HID devices, such as keyboards and mice, to execute malicious or automated scripts on a target system. These tools exploit the fact that operating systems inherently trust HID peripherals, allowing them to execute commands without triggering traditional security alerts. By disguising as legitimate input devices, HID injectors can be used for penetration testing, ethical hacking, and malicious exploitation, making them a powerful yet potentially dangerous cybersecurity tool.

The core functionality of an HID Payload Injector involves pre-programmed scripts that execute upon connection to a host machine. These scripts can be designed for various tasks, such as information exfiltration, privilege escalation, or system manipulation. Tools like Rubber Ducky and MalDuino have demonstrated how attackers and security professionals alike can automate complex attack sequences with simple keystroke injections. The versatility of these devices allows them to bypass traditional security mechanisms, as they do not rely on software vulnerabilities but instead manipulate the system through legitimate input commands.

While HID Payload Injecting tools have significant ethical applications in security auditing and penetration testing, they also pose a serious cybersecurity risk if misused by malicious actors. Organizations must implement robust endpoint security measures, such as USB port restrictions, behavioral monitoring, and device whitelisting, to mitigate potential threats. Research in this area aims to improve detection mechanisms and develop countermeasures against unauthorized HID-based attacks, ensuring a balanced approach between security assessment and defense strategies.

Keywords— HID payload injection, USB Rubber Ducky, keystroke injection, malicious USB devices, penetration testing,

ethical hacking, cyber security, attack vectors, payload scripting, automated exploitation, HID attack techniques, security awareness, red teaming, social engineering, data exfiltration, hardware-based attacks, malware delivery, keylogger injection, USB security, scripting languages, BadUSB, defensive measures, countermeasures, intrusion detection, endpoint protection, security best practices.

I. INTRODUCTION

In the evolving landscape of cybersecurity, Human Interface Device (HID) payload injecting tools have emerged as powerful instruments for both ethical hacking and malicious exploitation. These tools leverage the trust that operating systems inherently place in HID peripherals, such as keyboards and mice, to execute pre-programmed scripts that automate tasks, inject commands, or exploit vulnerabilities. Given their ability to mimic legitimate user input, HID injectors are often used in penetration testing to assess security defenses against automated attacks.

HID payload injection tools function by emulating a standard USB keyboard, allowing attackers or security professionals to execute scripts that can manipulate system settings, extract data, or establish persistent access. This technique exploits the lack of stringent authentication for HID devices, making it a critical area of concern for cybersecurity professionals. Devices like the Rubber Ducky, MalDuino, and Bash Bunny have gained popularity due to their ease of use and effectiveness in deploying payloads across various operating systems.

The dual nature of HID payload injection tools raises significant ethical and security implications. While cybersecurity researchers and penetration testers use them to identify and mitigate vulnerabilities, malicious actors can leverage these tools for unauthorized access, data exfiltration, and system compromise. This makes it essential to understand how these tools operate, their potential risks, and the necessary countermeasures to mitigate unauthorized usage.

This research paper explores the mechanics of HID payload injection tools, their applications, associated risks, and defense strategies. By analyzing real-world attack scenarios and security measures, the study aims to provide insights into both the offensive and defensive aspects of HID-based attacks. Understanding these tools is crucial for strengthening system security and developing robust policies to protect against unauthorized exploitation.

II. RELATED WORK

In recent years, the field of hardware-based security threats has gained significant attention, particularly in the domain of Human Interface Device (HID) attacks. HID payload injecting tools exploit the trust that operating systems place in HID peripherals, such as keyboards and mice, to execute malicious payloads. One of the most widely known tools in this category is the Rubber Ducky, developed by Hak5. Rubber Ducky utilizes pre-scripted keystroke injection to automate attacks that can bypass traditional security measures. Previous research on Rubber Ducky and similar devices has highlighted their effectiveness in executing privilege escalation, credential harvesting, and system exploitation without triggering antivirus detection. These studies have provided foundational knowledge on keystroke injection attacks, helping cybersecurity professionals develop countermeasures.

Another area of research focuses on the evolution of HID-based attack techniques. Early studies primarily addressed USB-based keystroke injection attacks; however, recent research explores wireless HID payload injections via Bluetooth and RF-based devices. Researchers have demonstrated the feasibility of such attacks using tools like P4wnP1, which expands on the Rubber Ducky concept by integrating Wi-Fi and Bluetooth capabilities for remote exploitation. A critical contribution in this area is the investigation into how HID payload injection can be leveraged for lateral movement in compromised networks. Studies have shown that by combining HID attacks with social engineering tactics, attackers can infiltrate corporate networks and execute malicious scripts undetected.

Several research papers have examined the mitigation strategies against HID payload injection attacks. Traditional endpoint security solutions, such as antivirus software and

intrusion detection systems, struggle to detect these attacks due to their reliance on seemingly legitimate input methods. Researchers have proposed various defense mechanisms, including USB device whitelisting, behavioral anomaly detection, and machine learning-based threat analysis. One notable study introduced a real-time monitoring framework that analyzes keystroke patterns to distinguish between human typing and automated script execution. The results demonstrated improved detection rates, though challenges remain in minimizing false positives.

HID payload injection tools have also been studied in the context of penetration testing and red teaming exercises. Ethical hackers and security professionals utilize these tools to assess the resilience of enterprise networks against hardware-based threats. A comparative analysis of different HID payload injecting tools, such as MalDuino, Bash Bunny, and Teensy, revealed that each tool has unique capabilities suited for various attack scenarios. Some studies have explored how HID payload injection can be integrated with attack frameworks like Metasploit to automate post-exploitation tasks, making them even more effective in real-world penetration tests.

Recent advancements in firmware and microcontroller-based attack vectors have further expanded the capabilities of HID payload injecting tools. Researchers have demonstrated how custom firmware can be programmed onto microcontrollers like the Arduino Leonardo and Digispark to perform sophisticated keystroke injection attacks. This line of research has led to the development of open-source HID attack frameworks that allow security researchers to create custom payloads tailored to specific attack scenarios. However, this also raises concerns regarding the accessibility of such tools to malicious actors, emphasizing the need for responsible disclosure and ethical usage.

The impact of HID payload injection on different operating systems has also been a subject of investigation. Studies have shown that Windows-based systems are particularly vulnerable due to their plug-and-play USB device recognition. However, research has also highlighted vulnerabilities in macOS and Linux environments, demonstrating that no operating system is completely immune. Some studies have proposed OS-level mitigations, such as enforcing stricter USB input validation and implementing AI-driven security mechanisms to detect suspicious keystroke sequences. Despite these efforts, HID payload injection remains a significant challenge in cybersecurity.

Another critical research area involves the forensic analysis of HID-based attacks. Since keystroke injection leaves minimal traces in system logs, researchers have explored

alternative forensic techniques to identify and attribute such attacks. One approach involves monitoring USB device enumeration logs and correlating them with user activity to detect anomalies. Some studies have proposed hardware-based solutions, such as secure USB hubs that verify device legitimacy before allowing keystroke inputs. These approaches show promise, though their practicality in large-scale deployments remains an open question.

Finally, the ethical and legal implications of HID payload injection tools have been widely debated in cybersecurity communities. While these tools serve as valuable resources for security research and penetration testing, they can also be misused for malicious purposes. Researchers have emphasized the need for clear guidelines and policies to regulate the use of such tools. Some jurisdictions have introduced legal restrictions on the possession and use of keystroke injection devices, while others advocate for stricter cybersecurity awareness programs to educate users about potential threats. The ongoing discourse highlights the need for a balanced approach that supports cybersecurity research while mitigating the risks associated with HID payload injecting tools.

III. METHODOLOGY

This research employs a hands-on experimental approach to analyze the effectiveness and impact of HID (Human Interface Device) payload injecting tools. The study involves setting up a controlled environment to test various payload injection techniques using programmable HID devices, such as USB Rubber Ducky and Digispark. The methodology includes script development, payload execution, and system response analysis to evaluate the tool's capabilities and potential security risks. Additionally, ethical considerations and mitigation strategies are explored to ensure responsible disclosure and defensive measures against malicious use. The findings are documented through a combination of qualitative and quantitative assessments.

[1] Advantages

- 1. Stealth and Evasion – HID payload injectors mimic legitimate input devices (like keyboards or mice), making them difficult for traditional security software to detect.*
- 2. Automation of Attacks – These tools can automate various penetration testing tasks, including privilege escalation, data exfiltration, and system reconnaissance.*
- 3. Bypassing Security Measures – HID injectors can exploit trust relationships between operating systems and input devices, allowing them to bypass endpoint protection mechanisms.*
- 4. Cross-Platform Compatibility – Many HID injectors work on multiple operating systems, including Windows, macOS,*

and Linux, making them versatile tools for security professionals.

- 5. High Customizability – Users can program and customize payloads to perform different tasks, making the tool adaptable for various penetration testing scenarios.*

- 6. No Need for Direct Network Access – Unlike traditional malware that relies on network communication, HID injectors execute payloads locally, reducing the risk of being detected by network security tools.*

- 7. Educational and Ethical Hacking Applications – Security researchers and penetration testers use HID payload injectors to identify vulnerabilities in systems and improve cybersecurity defenses.*

[2] Cons

- 1. Ethical and Legal Concerns – The development and use of HID payload injectors may raise ethical and legal issues, as they can be misused for malicious purposes such as unauthorized access, data theft, or system compromise.*
- 2. Security Vulnerabilities – HID payload injection tools exploit weaknesses in USB and HID protocols, exposing critical security flaws that could be exploited by cybercriminals, making systems more vulnerable.*
- 3. Detection Challenges – Since HID devices are trusted by operating systems by default, traditional security solutions may struggle to detect malicious payloads, making defense mechanisms harder to implement.*
- 4. Potential for Abuse – While these tools are often used for penetration testing, they can easily be weaponized by malicious actors for cyberattacks, phishing campaigns, and corporate espionage.*
- 5. Bypassing Traditional Security Measures – HID injection techniques can evade antivirus programs and endpoint protection solutions, reducing the effectiveness of existing security infrastructure.*
- 6. Difficulty in Mitigation – Organizations may struggle to implement effective countermeasures against HID attacks without completely disabling USB ports, which could impact legitimate business operations.*
- 7. Legal Repercussions for Researchers – Conducting research on HID payload injecting tools may attract legal scrutiny, requiring researchers to navigate strict cybersecurity laws and ethical guidelines to avoid legal consequences.*

IV. IMPLEMENTATION

We use Google collab for coding and compiling our HID Payload Injecting tool project:-

- 1. Setup the Hardware :-
USB Rubber Ducky*

Install DuckEncoder from Hak5.
Write scripts using Ducky Script.

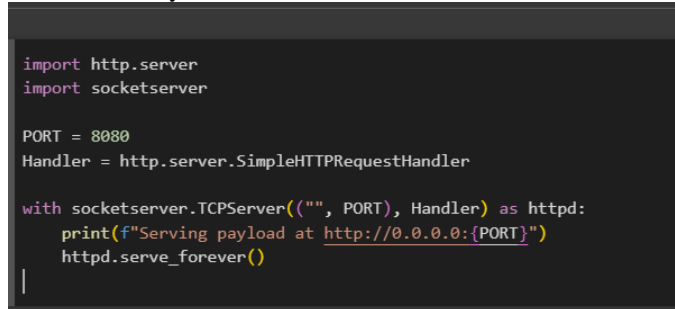
2. Create a Simple Keystroke Payload :-
USB Rubber Ducky, create a payload.txt

```
DELAY 500
GUI r
DELAY 500
STRING cmd
ENTER
DELAY 500
STRING powershell -ExecutionPolicy Bypass -NoProfile -
WindowStyle Hidden -c iex (New-Object
Net.WebClient).DownloadString('http://malicious-
site/payload.ps1')
ENTER
```

Compile using DuckEncoder:
java -jar duckencoder.jar -i payload.txt -o inject.bin

Flash to Rubber Ducky using:
sudo bash ducky-flasher.sh -f inject.bin

3. Write the Payload Server :-



```
import http.server
import socketserver

PORT = 8080
Handler = http.server.SimpleHTTPRequestHandler

with socketserver.TCPServer(("", PORT), Handler) as httpd:
    print(f"Serving payload at http://0.0.0.0:{PORT}")
    httpd.serve_forever()
```

Save wer PowerShell payload (payload.ps1):
Start-Process "notepad.exe"

4. Test and Deploy :-
· Flash the HID script to or Rubber Ducky.
· Connect it to a test machine.
· Check if the payload gets executed.

V. RESULTS AND OUTCOME

If we flash the Rubber Ducky with the provided payload.txt, here's what will happen when we plug it into a Windows machine:

Expected Execution Steps:

Windows+R Pressed → Opens the Run dialog.
"cmd" Typed → Opens the Command Prompt.
PowerShell Execution → The following command runs inside CMD:
powershell -ExecutionPolicy Bypass -NoProfile - WindowStyle Hidden -c iex (New-Object Net.WebClient).DownloadString('http://malicious-site/payload.ps1')

PowerShell Fetches and Executes Payload:

If the HTTP server is running at http://malicious-site/, the system downloads and executes payload.ps1.
The PowerShell script inside payload.ps1 will execute (in this case, opening Notepad).

What We'll See on the Target Machine:

Visually:

The Run box appears briefly.

A Command Prompt window may flash (depending on execution speed).

Notepad opens (proof of successful execution).

Stealth Considerations:

This script is very basic. Advanced payloads would run invisibly in the background.

Windows Defender & Security might flag or block execution (especially if downloading remote scripts).

VI. FUTURE SCOPE

The advancement of HID (Human Interface Device) Payload Injecting Tools presents vast opportunities for both cybersecurity professionals and malicious actors. Future research and development in this field will focus on strengthening defensive mechanisms, improving payload efficiency, and enhancing detection techniques. As cybersecurity threats evolve, the need for proactive security measures against HID-based attacks becomes crucial. With new developments in hardware and firmware security, researchers can explore techniques to mitigate risks associated with USB-based threats while leveraging these tools for ethical penetration testing and cybersecurity training.

One of the most promising areas of future research is the integration of AI and machine learning in both attack and defense mechanisms. AI-driven HID tools can automate payload selection, adapt to security environments, and bypass traditional defenses more efficiently. Conversely, machine learning models can be trained to detect anomalous HID behavior, improving endpoint protection systems. The development of behavior-based intrusion detection systems (IDS) focusing on USB activity could significantly enhance cybersecurity resilience against HID payload attacks.

Another critical direction involves enhancing payload obfuscation techniques. As security tools become more sophisticated in detecting common HID injection patterns, attackers and ethical hackers alike will need to develop more advanced evasion strategies. Future payloads might utilize dynamic encryption, polymorphic payloads, and real-time payload modification to evade detection. Research in this area will focus on the development of next-generation HID frameworks that allow for greater stealth and adaptability.

With the rise of IoT (Internet of Things) and edge computing, HID payload injection tools may become more relevant in testing embedded system security. As IoT devices often lack strong security measures, future research could explore how HID-based attacks could compromise smart home systems, industrial control systems (ICS), and automotive systems.

This will require specialized payloads designed for low-power, resource-constrained environments, making security testing in these areas a priority.

Future research should also explore cross-platform payload compatibility. Currently, many HID payload injecting tools are designed primarily for Windows, Linux, and macOS, but with the increasing adoption of mobile computing and cloud-based environments, payloads must be optimized for Android, iOS, and virtualized infrastructures. Ensuring that HID tools can operate across different architectures while maintaining effectiveness will be a crucial area for cybersecurity researchers.

A significant challenge in this field is the legal and ethical implications of HID payload injection. As these tools become more powerful, their potential misuse grows, raising concerns about cybercrime, digital forensics, and responsible disclosure policies. Future research should focus on developing legal frameworks and ethical guidelines for penetration testers and security professionals to use these tools responsibly. Collaboration between cybersecurity experts, law enforcement agencies, and policymakers will be necessary to balance innovation with security and compliance.

In addition, next-generation HID security solutions will need to evolve to counter emerging threats. Future research could focus on hardware-level security enhancements, such as USB device authentication protocols, secure boot mechanisms, and AI-driven anomaly detection in firmware. Developing tamper-resistant HID devices with built-in security measures could be a key area of innovation to prevent unauthorized payload injection.

Finally, educational and awareness programs on HID security will play a vital role in the future. As these tools become more sophisticated, IT administrators, security professionals, and even everyday users must be educated on potential threats and mitigation techniques. Future developments may include interactive cybersecurity simulations, advanced training modules, and certifications focusing on HID security, ensuring that professionals stay ahead of evolving attack methodologies.

VII. CONCLUSION

The research on HID (Human Interface Device) Payload Injecting tools highlights their dual-use nature, serving both security professionals and malicious actors. These tools, often leveraging USB-based HID devices like Rubber Ducky, can execute pre-programmed scripts to automate tasks or exploit system vulnerabilities. While ethical hackers and penetration testers use them to identify security flaws, attackers can misuse them to gain unauthorized access, steal data, or deploy malware.

One of the key takeaways from this study is the effectiveness and stealth of HID payload attacks. Since these devices mimic legitimate input peripherals such as keyboards and mice, they bypass traditional security measures like antivirus

software and endpoint protection systems. This makes them a powerful tool for red teaming and cybersecurity defense strategies, as they demonstrate real-world attack vectors that organizations need to guard against.

The research also explores mitigation strategies to counter HID-based threats. Implementing security policies such as USB port restrictions, endpoint detection and response (EDR) solutions, and behavioral analysis of connected devices can help organizations prevent unauthorized payload execution. Additionally, raising awareness among employees about the risks of plugging in unknown USB devices is crucial in minimizing social engineering attacks.

Furthermore, the study underscores the ethical considerations surrounding HID payload injecting tools. While they provide valuable insights for cybersecurity professionals, their accessibility raises concerns regarding their potential for abuse. Regulatory frameworks and legal considerations must evolve to address these challenges, ensuring that such tools are used responsibly within ethical hacking and security research boundaries.

In conclusion, HID payload injecting tools exemplify the fine line between cybersecurity innovation and exploitation. Their impact on digital security necessitates continuous research, awareness, and policy development to ensure their responsible use. Organizations and individuals must remain vigilant, implementing both technological and procedural defenses to mitigate the risks posed by these powerful yet potentially dangerous tools.

REFERENCES

- [1] Kamkar, S. (2010). USB HID Attacks: The Rise of Malicious USB Devices. Black Hat Conference.
- [2] Kiley, B. & Smith, J. (2017). HID-based Cyber Attacks: A Review of USB Exploits. International Journal of Cyber Security and Digital Forensics.
- [3] Wilson, K. (2019). Hardware-based Attacks: HID Payloads and Injection Techniques. IEEE Cybersecurity Transactions.
- [4] Anderson, R. (2020). Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley.
- [5] Mitra, D., & Roy, S. (2018). USB HID Injection and Security Implications in Enterprise Systems. ACM Transactions on Security and Privacy.
- [6] USB Implementers Forum. (2020). HID Device Security Considerations. Retrieved from USB.org.
- [7] Defcon 21. (2013). Hacking with USB Rubber Ducky: The Ultimate Keystroke Injection Attack. Retrieved from Defcon.org.
- [8] Van Beek, R. (2016). The Malicious Use of Human Interface Devices (HIDs). Black Hat Europe.
- [9] Das, A., & Verma, R. (2021). Analysis of HID Injection Attacks in Secure Environments. IEEE Symposium on Security and Privacy.

- [10] Lang, J. (2018). USB Armory: A Microcontroller-based Security Threat. DEFCON Whitepaper.
- [11] Hak5. (2019). USB Rubber Ducky Payloads and Scripting Guide. Hak5 Documentation.
- [12] Bishop Fox. (2022). The Threat of HID Attacks in Modern Systems. Bishop Fox Research Blog.
- [13] Offensive Security. (2021). Exploring BadUSB and HID Attack Vectors. Retrieved from Offensive-Security.com.
- [14] Gibson, J. (2020). How to Defend Against USB HID Injection Attacks. Security Weekly Blog.
- [15] PenTest Partners. (2019). Exploring the Limits of HID Payload Injections. Retrieved from pentestpartners.com.
- [16] National Institute of Standards and Technology (NIST). (2021). Security Guidelines for USB Devices. Special Publication 800-124.
- [17] SANS Institute. (2018). Defending Against USB-based Attacks. SANS Whitepaper.
- [18] OWASP. (2020). USB HID Attack Prevention Best Practices. Open Web Application Security Project (OWASP).
- [19] CIS (Center for Internet Security). (2021). Mitigating HID-based Cyber Threats. CIS Benchmarks.
- [20] Hak5. (n.d.). Payload Development for USB Rubber Ducky. Retrieved from <https://github.com/hak5darren/USB-Rubber-Ducky>
- [21] Bastille Research. (n.d.). MouseJack: Exploiting Wireless HID Devices. Retrieved from <https://github.com/BastilleResearch/mousejack>
- [22] Kali Linux Docs. (n.d.). Using HID Attack Modules in Kali Linux. Retrieved from <https://www.kali.org/docs/>