IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 12, December 2024

DOI: 10.17148/IJARCCE.2024.131231

Exploiting Vulnerabilities using Keystroke Injections

Mr. Dhanraj[1], Varun Hegde[2], Smayan C N[3]

Dept of CSE-Cyber Security, RNSIT, Bengaluru, India[1-3]

Abstract: In this paper, we discuss the development and implementation of a malicious and non-malicious payload delivery system using the Digispark microcontroller. Our system leverages extremely fast, automated keystroke injections to deliver a variety of payloads. We demonstrate how these injections enable the execution of a reverse PowerShell shell, establishing a remote connection for command execution. Additionally, we present a payload that stealthily retrieves Wi-Fi credentials, creates a backdoor for persistent access, and escalates privileges to gain full control of the target system. Our approach optimizes keystroke injection for speed, bypassing traditional security measures by simulating human input, thereby minimizing detection. We also explore ethical applications, such as penetration testing, and highlight the dual-use nature of the Digispark-based delivery system. Through this work, we contribute novel methods of leveraging the Digispark microcontroller for both malicious and ethical purposes, advancing the understanding of microcontroller-based payload delivery in cybersecurity.

Index Terms: Digispark, keystroke injection, reverse PowerShell shell, Wi-Fi password theft, backdoor, privilege escalation, payload delivery, cybersecurity.

I. INTRODUCTION

A microcontroller is a small, self-contained circuit board that can be programmed to control and interact with other devices. In this project, we use the Digispark microcontroller, an Arduino-compatible development board known for its compact size and versatility. The Digispark is equipped with an ATtiny85 chip, which includes a processor, memory, and input/output pins that allow it to interface with external systems and simulate keystrokes on a victim's machine. Despite its small form factor, the Digispark is powerful enough to execute complex payloads, making it an ideal tool for delivering attacks through automatic keystroke injection. A payload is a piece of code or script designed to perform specific actions once it's executed on a target system.

In this case, the payload is scripted to carry out several functions, such as establishing a reverse shell, stealing Wi-Fi credentials, creating a persistent backdoor, and escalating privileges on the compromised machine. The delivery system works by plugging the Digispark microcontroller into the victim's machine via a USB port. Upon connection, the microcontroller mimics the actions of a keyboard, injecting rapid keystrokes that execute the payload. The keystrokes trigger the script to run, starting with the reverse shell, followed by retrieving network information, setting up the backdoor, and escalating system privileges.

This reverse shell enables remote control of the victim's machine from a Linux-based attacker machine, allowing the attacker to execute commands remotely. This project highlights the speed and effectiveness of the Digispark in delivering payloads through keystroke injection, demonstrating both its malicious capabilities and ethical use cases, such as penetration testing, where it can help identify and address security vulnerabilities. [1]

II. LITERATURE SURVEY

Several papers have explored the use of microcontrollers and USB-based attacks for various purposes, including security testing and assistive technologies.

- M. A. Moore, "Human Interface Device (HID) Attacks: A Review" (2022): This paper reviews HID-based attacks and how USB devices, like keyboards, can be exploited for payload delivery. We extend this work by demonstrating the use of the Digispark microcontroller for payload delivery through keystroke injection, introducing a new attack vector

- T. K. Bell, R. M. Arnold, and F. M. Jennings, "Assistive Technologies Using Microcontrollers for Disabled Individuals" (2020): This paper explores the use of microcontrollers in assistive technologies. We build on this versatility by applying microcontrollers, specifically the Digispark, for cybersecurity attacks, highlighting both their beneficial and malicious potential.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 12, December 2024
DOI: 10.17148/IJARCCE.2024.131231

- R. K. Patel, ”Ethical Hacking: A Guide to Practical Penetration Testing Techniques” (2021): Patel’s work covers penetration testing and ethical hacking tools. We extend this by using a small, affordable microcontroller for penetration testing, introducing new methods for system exploitation.

- S. P. Sullivan and R. C. Frye, ”USB-based Malware and Its Impact on Industrial Control Systems” (2019): This paper addresses USB-based malware in industrial systems. We expand on their findings by demonstrating how the Digispark microcontroller can be used for both malicious and ethical cybersecurity attacks.

- J. D. Shaw and S. B. Lee, ”Automating Network Configuration and Security Testing Using Microcontrollerbased Systems” (2022): Shaw and Lee focus on automating network configuration and security testing with microcontrollers. We build on this by showcasing a microcontroller-based payload delivery system for exploiting vulnerabilities and achieving privilege escalation.

Our paper contributes new knowledge to the field by focusing on a practical implementation of the Digispark microcontroller for payload delivery through keystroke injection, including a novel combination of reverse shell execution, WiFi password theft, backdoor creation, and privilege escalation.

The work emphasizes both the potential dangers of such technology when misused and its ethical applications in penetration testing and cybersecurity defense.

III. HUMAN INTERFACE DEVICE (HID) ATTACKS

Human Interface Device (HID) attacks leverage peripherals, such as keyboards and mice, that are commonly recognized by operating systems as trusted input devices. Microcontrollers like the Digispark, which can emulate these HID devices, are exploited to inject malicious payloads into a target system.

HID attacks are particularly effective because they exploit the system’s natural trust in input devices and the assumption that only human input will occur. This section explores the nature of HID attacks, how they work, and the specific techniques involved in payload delivery through microcontroller- based HID emulation.

A. Understanding HID Attacks

HID attacks involve the use of a device (often a microcontroller, such as Digispark or Arduino) that masquerades as a keyboard or mouse. When plugged into a victim machine, these devices can send keystrokes or mouse movements that are interpreted as legitimate user input. The microcontroller is typically programmed to execute commands on the system in a rapid, automated manner.

This process allows an attacker to execute malicious payloads without physical interaction from the user. HID attacks are particularly insidious because they occur at the hardware level, often bypassing traditional softwarebased security measures such as antivirus software, firewalls, or intrusion detection systems. As a result, HID attacks are effective against both well-secured machines and unpatched systems.

B. How HID Attacks Work

The key to HID attacks is the ability of microcontrollers to emulate HID devices. The attack typically unfolds as follows:

- Connection to the Target System: The microcontroller (e.g., Digispark) is plugged into a USB port on the victim’s computer. The system detects the device as a standard HID, such as a keyboard or mouse.
- Keystroke Injection: Once the device is connected, the microcontroller injects a series of pre-programmed keystrokes into the system. These keystrokes can be used to open applications, run scripts, or download additional malicious payloads.
- Payload Execution: The keystrokes trigger the execution of the malicious payload, which could include creating a reverse shell, stealing sensitive data (e.g., WiFi credentials), or escalating privileges on the system.
- Persistence and Backdoor Creation: Some HID attacks are designed to install a persistent backdoor, allowing the attacker continued access to the system even after the initial payload is executed. This can be achieved through the injection of a malicious script that remains active after the system reboots.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 12, December 2024

DOI: 10.17148/IJARCCE.2024.131231

C. Payloads Scripted by Us Delivered via HID Attacks

HID attacks are often used to deliver payloads with various malicious objectives, including:

- Reverse Shell: A reverse shell allows an attacker to remotely control the victim's system by establishing an outbound connection to the attacker's machine. This enables command execution and data exfiltration.
- Wi-Fi Credential Theft: Some HID-based payloads are designed to retrieve and send saved Wi-Fi credentials from the target machine to the attacker.
- Privilege Escalation: Microcontroller-based HID attacks can escalate privileges, enabling the attacker to gain administrative access to the victim's system.
- Backdoor Installation: A backdoor is installed to ensure that the attacker can re-enter the system without the need for physical access or repeated exploitation of vulnerabilities. [2]

D. Advantages of HID Attacks

- Bypassing Security Software: Since HID attacks exploit the system's trust in input devices, they can bypass most traditional security mechanisms like antivirus programs or network firewalls.
- Speed and Efficiency: HID attacks are fast and effective. A microcontroller can rapidly inject keystrokes, completing the execution of complex payloads in a matter of seconds.
- No User Interaction Required: Unlike phishing attacks or social engineering, HID attacks require minimal interaction from the victim. The attacker can inject payloads automatically, without the user needing to click on a malicious link or open an infected file.
- Stealth: HID attacks can be stealthy, as they mimic human input and do not generate the typical signs of an intrusion, such as abnormal network traffic or unusual file system activity.

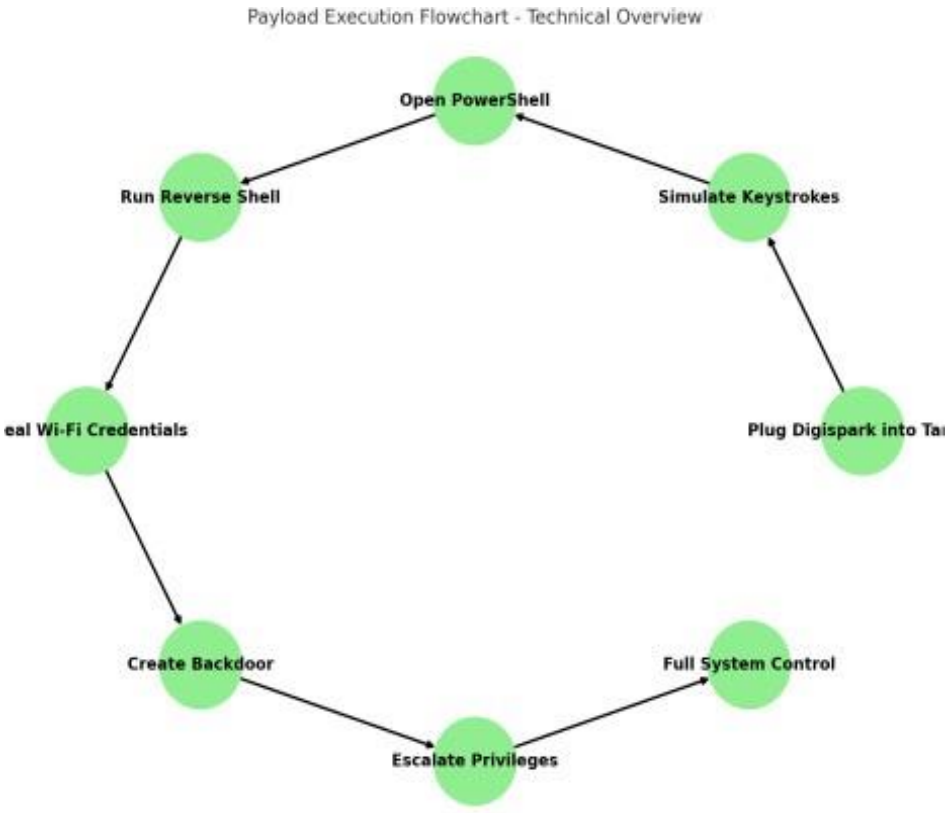

Fig. 1. Flowchart of the Payload Execution Process

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 12, December 2024

DOI: 10.17148/IJARCCE.2024.131231

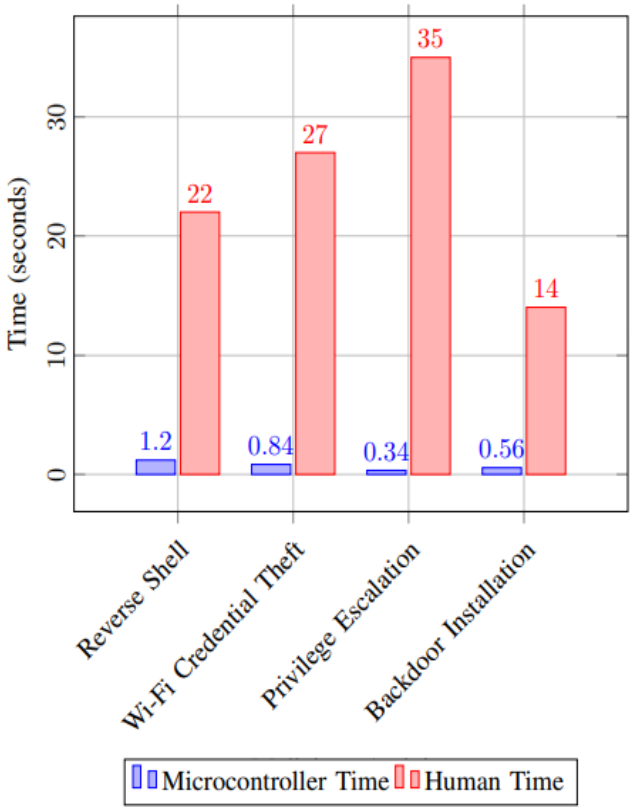

Fig. 2. Time taken for various malicious activities using a microcontroller versus human execution

IV. MITIGATION TECHNIQUES FOR HID-BASED ATTACKS

As the threat of HID-based attacks, such as those involving keystroke injection through microcontrollers like the Digispark, continues to grow, implementing effective countermeasures is essential to protect systems and networks. Below are several mitigation techniques that can significantly reduce the risk of such attacks:

A. USB Port Control and Restrictions

- USB devices, being the primary conduit for HID-based attacks, need to be tightly controlled. The following strategies can be implemented to mitigate the risk of HID device exploitation:
- Disabling USB Ports: One of the most effective ways to mitigate HID attacks is to physically disable unused USB ports or disable USB ports in the system BIOS/UEFI settings. This prevents unauthorized devices from connecting to the system.
- Using USB Data Blockers: USB data blockers can be used to prevent data transmission through USB ports while still allowing the device to be powered. This is particularly useful when charging devices or when a system needs to be protected from malicious input.
- USB Port Monitoring Software: Installing software that monitors all USB devices connected to the system can help detect unauthorized devices. Alerts can be triggered if a non-whitelisted device is plugged in, allowing for immediate action.

V. EMERGING TRENDS AND FUTURE IMPLICATIONS

As we explore the growing use of microcontroller-based payload delivery systems, several emerging trends are becoming evident, signaling the increasing sophistication and potential applications of these devices in both malicious and ethical contexts.

The future of microcontroller-based attacks, specifically using systems like the Digispark, holds various implications for cybersecurity and penetration testing.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 12, December 2024
DOI: 10.17148/IJARCCE.2024.131231

A. Advancements in Microcontroller Capabilities

We foresee significant advancements in microcontroller capabilities, particularly in terms of processing power, connectivity, and size. With the integration of Wi-Fi and Bluetooth modules into compact boards, we could see more seamless and rapid delivery of payloads, expanding beyond USBbased HID attacks. This opens up the possibility of launching attacks over the air (OTA), making the delivery system even harder to detect and mitigate. The continued miniaturization of microcontrollers will also allow for more discrete and covert payload delivery methods.

B. Increased Use of Artificial Intelligence in Payload Delivery

Artificial intelligence (AI) and machine learning (ML) could play a role in enhancing the stealth and efficiency of microcontroller-based attacks. For instance, AI could be used to dynamically adapt the behavior of the payload to evade detection by security software. Machine learning models could analyze the target environment and adjust the keystroke injection patterns to avoid triggering security alerts, making the payload delivery process even more sophisticated. Additionally, AI-driven payloads could be more effective at identifying vulnerabilities and exploiting them in real-time, which could dramatically increase the success rate of such attacks.

C. The Rise of Autonomous Payload Delivery Systems

We also anticipate the development of fully autonomous payload delivery systems, where the microcontroller is not only responsible for executing the attack but is also capable of learning and adapting to bypass security mechanisms without human intervention. These systems would integrate with larger botnets or networks of compromised devices, enabling them to scale their attacks and increase their effectiveness. This trend may lead to more widespread, decentralized attacks, making it harder to trace and counteract these cyber threats.

D. Ethical Penetration Testing and Security Research

On the ethical side, we foresee the growing use of microcontroller-based payload delivery systems in penetration testing and security research. As these devices become more accessible and affordable, they can be integrated into security audits to help organizations identify vulnerabilities in their systems. However, with these advancements, we must also consider the ethical implications of using such tools. We believe that the cybersecurity community will need to implement stronger guidelines and regulations to ensure that these technologies are used responsibly and only for authorized testing and security purposes.

E. Countermeasures and Defenses

In response to these emerging trends, we expect that defense mechanisms will evolve as well. Traditional security systems such as antivirus programs and firewalls may become less effective against HID-based attacks, requiring the development of new, more advanced security solutions. Future defense mechanisms could include USB filtering systems, AI-powered anomaly detection, and behavior-based security models that can identify suspicious input patterns, regardless of the device initiating them. In conclusion, while the capabilities of microcontroller based payload delivery systems continue to expand, so too do the challenges and risks they pose.

F. Device Whitelisting and Access Control

By restricting which devices can be connected to a system, device whitelisting helps prevent unauthorized HID devices from interacting with the system.

- Device Whitelisting: Systems can be configured to only recognize trusted USB devices. This involves creating a list of approved devices, such as keyboards, mice, or USB flash drives, and blocking all others from connecting. Many endpoint management systems allow for such configuration.

- Access Control Lists (ACLs): ACLs can be used to define who has access to USB ports and which devices are permitted. With appropriate settings, administrators can prevent unauthorized users from plugging in potentially malicious devices.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 12, December 2024
DOI: 10.17148/IJARCCE.2024.131231

G. Operating System and Firmware Hardening

Operating systems and firmware play a crucial role in defending against HID-based attacks. By implementing certain hardening techniques, vulnerabilities that can be exploited by HID attacks can be minimized. • Disabling HID Devices in OS Settings: Some operating systems provide the option to disable HID devices altogether or limit their functionality to trusted sources only. Disabling USB keyboards and mice from non-trusted sources helps mitigate HID-based attacks.

- Regular Software Updates and Patch Management: Keeping the operating system, device drivers, and firmware up to date is crucial in preventing exploitation of known vulnerabilities. Ensuring that patches for USBrelated vulnerabilities are applied promptly can protect against attacks like Stuxnet or Digispark-based keystroke injection.

- Trusted Platform Module (TPM) and Secure Boot: Enabling TPM and Secure Boot features can prevent unauthorized code execution during the boot process, which is often the stage at which HID-based attacks may initiate.

H. USB Monitoring and Behavior Analysis

Behavioral analysis tools can help detect unusual activity that might be indicative of an HID-based attack. These tools focus o I. Real-World Applications and Implications While HID attacks are often associated with malicious actors, they also have ethical use cases in penetration testing. Security professionals can use HID emulation to simulate attacks and identify vulnerabilities in a system's defenses.

By testing how a system responds to keystroke injection and unauthorized input, organizations can better secure their systems against such attacks. In summary, HID attacks are a powerful method of delivering payloads through microcontrollers like the Digispark, exploiting the system's trust in USB input devices. They are fast, stealthy, and effective, making them a significant concern for cybersecurity. Understanding how these attacks work and implementing countermeasures is critical for defending against them. [3]

I. Vulnerabilities Exploited

- USB-based Attacks: USB devices serve as a common vector for delivering payloads. In both Stuxnet and Digisparkbased HID attacks, malicious code is injected through USB interfaces, bypassing network security and exploiting system trust in connected peripherals.
- Unpatched Software Vulnerabilities: Exploiting unpatched vulnerabilities in operating systems and software remains a critical attack vector. Both Stuxnet and Digispark payloads rely on unaddressed security flaws, whether in Windows or device drivers, to gain unauthorized access.
- Social Engineering and User Trust: HID-based attacks, like those involving Digispark, leverage the system's inherent trust in input devices (e.g., keyboards) to execute commands without user awareness, similar to how Stuxnet capitalized on human trust for initial infection.
- Weak Security Configurations: Weak security configurations, such as improper access controls and privilege settings, are commonly exploited in both Stuxnet and HID-based payloads. These vulnerabilities enable privilege escalation and facilitate the execution of malicious payloads.

VI. CASE STUDY: STUXNET

A. Stuxnet Overview

Stuxnet, discovered in 2010, was a highly sophisticated malware targeting industrial control systems. It spread through USB devices and exploited zero-day vulnerabilities in Windows and Siemens software to sabotage Programmable Logic Controllers without detection.

B. Stuxnet and Digispark Payload Parallels

Both Stuxnet and Digispark payloads use USB devices to infiltrate systems and bypass traditional network defenses. Stuxnet employed complex, multi-stage exploits targeting critical infrastructure, while Digispark payloads use HID emulation for rapid, automated keystroke injections to deliver payloads like reverse shells and data theft.

C. Lessons and Implications

Stuxnet's use of USB as an attack vector underscores the potential for portable devices to exploit vulnerabilities. Similarly, Digispark-based HID attacks highlight the risks of untrusted USB devices used for payload delivery, making such attacks effective against both industrial and standard computing systems.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 12, December 2024

DOI: 10.17148/IJARCCE.2024.131231

VII. ETHICAL USE CASES FOR KEYSTROKE INJECTIONS

Keystroke injection attacks, where a device such as a USB microcontroller (e.g., Digispark or Arduino) simulates keyboard input to control a target system, can be used in ethical scenarios to automate tasks and improve security.

Below are some ethical use cases for keystroke injection:

A. Automating Repetitive Tasks

- Workflow Automation: Keystroke injection can be used to automate repetitive tasks that involve entering predefined commands, passwords, or settings. For example, an automation system could use keystroke injection to fill out forms, configure software, or execute common administrative tasks on computers.
- Text Expansion: Custom keystroke injection scripts can be created to automate common phrases or email responses, reducing typing time for frequently used phrases or templates.

B. Educational Tools for Teaching Programming

- Simulated Coding Challenges: Keystroke injection can be used to simulate typing patterns in programming environments during educational workshops or competitions, helping instructors demonstrate real-time coding techniques and system configurations.
- Programming Demonstrations: Instructors can use keystroke injection to showcase specific actions in a programming environment, allowing students to follow along without manual input.

C. Accessibility Solutions

- Assistive Technology for People with Disabilities: Keystroke injection devices can be used as assistive tools for individuals with disabilities who may find it difficult to use traditional input devices. For instance, custom keystroke injection setups can enable users to type or interact with their devices through simplified or alternative inputs.
- Customizable Input Methods: For users who have difficulty typing on a traditional keyboard, a keystroke injection device can provide a specialized solution, such as sending specific keystrokes with a single button press. [4]

D. Software Testing and Quality Assurance

- Automated Software Testing: Developers and QA testers can use keystroke injection to automate application testing, simulating user input to ensure software behaves as expected.
- Simulating User Interactions: Keystroke injection can be used during usability tests to simulate user inputs, allowing developers to gather data on application responsiveness.

E. Security Audits and Penetration Testing

- Penetration Testing (Controlled Environment): Ethical hackers can use keystroke injection in a controlled environment to test system vulnerabilities, assessing whether systems are protected against unauthorized input from USB-connected devices.
- Testing Anti-Malware Software: Keystroke injection can be used to simulate keyloggers or other input-based attacks to test the effectiveness of security software in detecting and preventing malicious behavior.

F. IT Administration and Automation

- Bulk Configuration Changes: IT administrators can use keystroke injection to automate system configuration across multiple computers, saving time during large-scale deployments.
- System Updates and Patching: Keystroke injection can help automate the process of entering commands during system updates or patching procedures, reducing manual intervention.

G. Demonstrations in Security Awareness Training

- Simulated Attacks for Awareness Training: Keystroke injection can be used in security training programs to demonstrate how easily attackers can exploit USB devices, highlighting the importance of securing USB ports.
- Creating Controlled Vulnerabilities: Ethical security professionals can use keystroke injection as part of simulated attack scenarios to educate users on recognizing and responding to potential threats.

H. Custom Hotkeys and Productivity Enhancements

- Creating Custom Keyboard Shortcuts: Keystroke injection can be used to program custom keyboard shortcuts or macro commands that enhance productivity by executing frequently used sequences with minimal effort.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 12, December 2024
DOI: 10.17148/IJARCCE.2024.131231

- Media Control Systems: Professionals using multimedia software can leverage keystroke injection to create hotkeys that control media playback and other functionalities, streamlining their workflow.

I. User Authentication Automation

- Automated Login: Keystroke injection can be used to automate typing of credentials, improving efficiency when accessing systems with repetitive login requirements.
- Multi-Factor Authentication Simulations: Keystroke injection can simulate the entry of authentication codes as part of a multi-step verification process.

J. Compliance and Security Audits

- USB Security Auditing Tools: Keystroke injection can be used to test USB security policies by simulating how an attacker might use an USB device to gain unauthorized control.
- Access Control Testing: Keystroke injection can be used to evaluate whether a system's access control mechanisms effectively prevent unauthorized users from executing commands via USB inputs. [5]

VIII. CONCLUSION

In this paper, we explored the use of microcontroller-based Human Interface Device (HID) attacks for payload delivery, demonstrating both malicious and ethical applications. By leveraging the Digispark microcontroller, we were able to simulate keystroke injection attacks, reverse shells, privilege escalation, and data exfiltration, highlighting the potential risks associated with USB-based attacks.

We also discussed the ethical applications of these techniques in penetration testing, assistive technologies, and network configuration testing, emphasizing their usefulness in both educational and practical environments.

The research underscores the importance of robust cybersecurity measures to prevent HID-based attacks and improve the resilience of network systems. Future work can focus on improving detection mechanisms and developing countermeasures against such attacks, ensuring a balance between security and functionality in modern systems.

ACKNOWLEDGMENT

We would like to thank **Dr. Rajkumar, Dr. Kiran P, Dr. Dhanraj, and Mrs. Yashasvini MN** for their guidance and support throughout this project. Our gratitude also goes to the entire department of CSE_Cyber Security at RNSIT for their continuous encouragement and resources.

REFERENCES

[1] R. K. Patel, Ethical Hacking: A Guide to Practical Penetration Testing Techniques. 2021.
[2] J. D. Shaw and S. B. Lee, "Automating network configuration and security testing using microcontroller-based systems," 2022.
[3] M. A. Moore, "Human interface device (hid) attacks: A review," 2022.
[4] T. K. Bell, R. M. Arnold, and F. M. Jennings, "Assistive technologies using microcontrollers for disabled individuals," 2020.
[5] S. P. Sullivan and R. C. Frye, "Usb-based malware and its impact on industrial control systems," 2019.