iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.51045

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Bad USB as HID and it's Mitigations

Sankar E[1], Megha Shyam Raju S[2], Sheshank Reddy K[3]
Computer Science and Engineering Department, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya Deemed University, Kanchipuram, Tamil Nadu, India

Abstract: ***A bug known as "Bad USB" allows a hacker to manipulate a USB flash drive's firmware. Malicious code is automatically activated when the Bad USB device is connected into the USB port of the host system. It is challenging to identify the malicious code since the host system interprets the harmful activities as legitimate ones required to load the USB device. Additionally, antivirus software is unable to detect firmware that has been modified because it examines the storage region rather than the firmware area. The vulnerability has a detrimental knock-on impact since a large number of computer peripherals (similar to USB flash drives, keyboards) are connected to the host machine using the USB protocols. Universal Serial Bus (USB) has become the primary connecting port for cutting-edge computers as a result of its universality. Programmability makes it easier for operating system and tackle manufacturers to develop their goods and associated firmware, but as of right now, a fix for the issue is unknown. In this article, we analyze the tampered area of the firmware that occurs when a good USB device is switched out with a bad one and provide a fix to ensure the integrity of the area when the USB thrills***
Keywords: USB, HID, Raspberry pi Pico, Mitigations, Payload.

I. INTRODUCTION

The Universal Serial Bus, or USB, was created in the middle of the 1990s to standardize the bus for consumer electronics connections.
In terms of speed, power capacity, durability, physical size, and compatibility, USB has an edge over prior interface standards like parallel ports. Basic communication features of USB allow its controller chips in peripherals to be reprogrammed in order to switch from one kind to another. There is no defense against such reprogramming in the USB controller chips, including those in USB flash drives.
As a result, the firmware in all incarnations in USB peripherals can be modified to hide the attack code, which raises numerous possible security concerns.

II. LITERATURE SURVEY

Traditional defense mechanisms cannot stop attacks like the one represented by Bad USB, but the majority of antivirus programmes can spot the introduction of malware via a USB stick. Traditional defense mechanisms are unable to recognize attacks of the kind represented by Bad USB; the majority of antivirus programmes can only detect the introduction of malware through a USB stick but are unable to access the firmware or determine whether it has been altered.
The proposed System consists of two procedures. The first procedure is to corroborate the integrity of the area which should be fixed indeed if the firmware is streamlined.
The verification system uses hashes, and the target area includes descriptors. The alternate procedure is to corroborate the integrity of the changeable area when the firmware is streamlined. The verification system use law signing, and the target area includes the function area of the firmware.
As a result, they provide in this paper a method to check the reliability of the firmware or driver that BAD USB installs. This technique can be used to develop preventative measures against malicious BAD USB behaviors.
Because a lot of computer peripherals (such as USB flash drive, keyboard) are connected to host system with the USB protocols, the vulnerability has a negative ripple effect. However, the countermeasure against the vulnerability is not known now. In this paper, we analyze the tampered area of the firmware when a normal USB device is changed to the Bad USB device and propose the countermeasure to verify the integrity of the area when the USB boots.

III. PROBLEM STATEMENT

Recent rapid technological advancements have made the USB an appealing security hot zone. For many years, the phrase "Bad USB" has also been used to describe a major problem with such USB peripherals apps. Since the USB may be reprogrammed, as was previously mentioned, an attack could be launched by simply connecting a USB device to a computer. From this vantage point, Bad USB is essentially described as a kind of firmware hack for USB devices.

©IJRASET: All Rights are Reserved

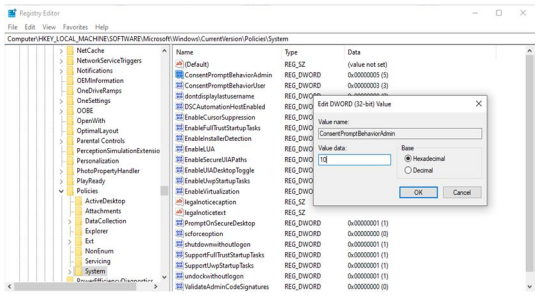
Once the Bad USB virus has been installed, USB devices are capable of simulating a keyboard and a logged-in user. For instance, a Plug-and-play USB can simulate keyboard or mouse input to instantly start the secret attack code when connected to a computer. Additionally, the infected USB device has the ability to mimic the network card and change the DNS configuration to reroute data. Malware behavioral detection is typically challenging to do because malware scanners cannot access the firmware in USB peripherals and the infected USB device can pretend to be a fresh device that the user has just casually attached. However, it is simple to get around such safeguards. In contrast to Bad USB, which was infected with malicious software, BadUSB2 is a newer version of Bad USB that is capable of a man-in-the-middle assault.

IV. SCOPE OF PROJECT

The scope of this project is to create a Bad USB using a Raspberry Pi Pico microcontroller that would resemble a USB like a flash drive so that when a user plugs it into the system, it may send keystrokes so quickly that the user has no idea what is going on. The attacker can acquire what he wants without ever touching the keyboard by sending keystrokes, which are contained in the payload. By demonstrating how it functions, we can learn how to defend in the first place.

V. PROPOSED METHOD

We can alter a policy through the registry editor to prevent Bad USBs from obtaining administrator capabilities and messing with the system, as indicated in the photo, even though the existing ways are straightforward but object other things and make it difficult for users to access USB ports. In order for the system to prompt for a password rather than just a yes or no response when the USB asks for administrator privilege, the user must modify the number from 5 to 1. The system user can access the privileges because the Bad USB doesn't have a password in the script, but it cannot. The group administrator can adjust this much more successfully using group policies.



VI. ARCHITECTURE

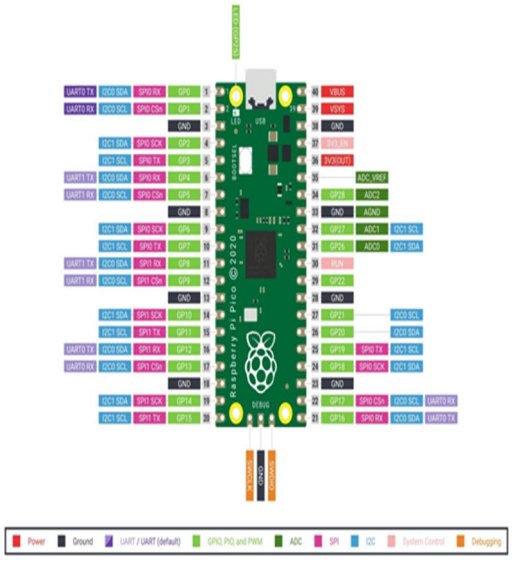


Fig.1 Architecture

VII. IMPLEMENTATION

The Raspberry Pi Pico board has to have the Circuit Python firmware uploaded to it. Keystrokes must be added to the user's system. User must design the payload in accordance with the specifications. Insert the Python code into the Raspberry Pi Pico via upload. User must design the payload in accordance with the specifications. Put the Raspberry Pi Pico into the USB port of the computer and let the code run its course.

VIII. OUTPUT



Fig.2.1 The raspberry pi pico connected to a system with the required code

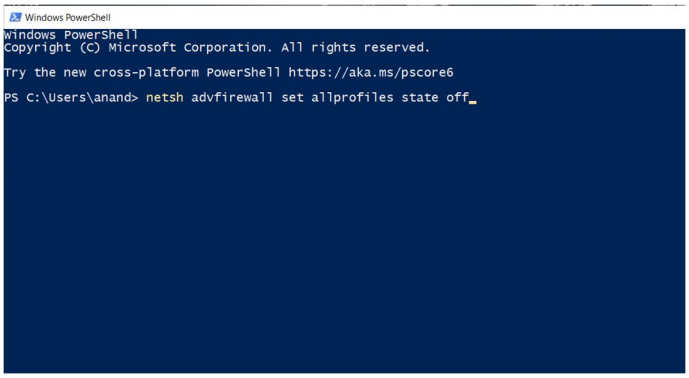


Fig.2.2 After inserting raspberry pi pico the windows firewall has been turned off.

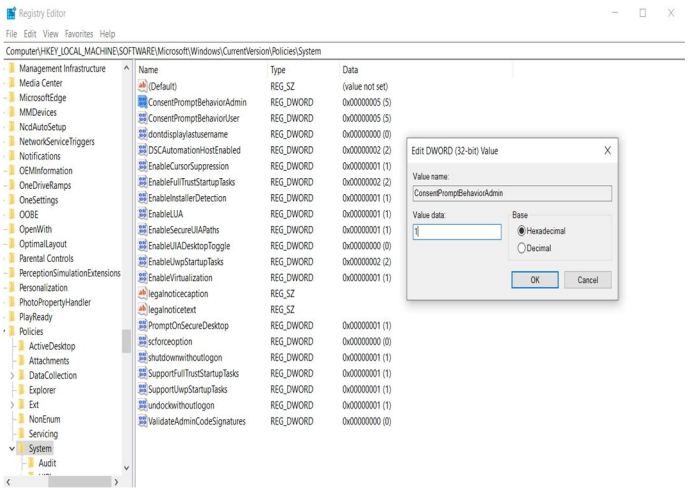


Fig.2.2 If we set the value the value to 1, we can secure the USB port area.

IX. CONCLUSION

Real-world users are very concerned about USB security. We presented a novel method to defend against Bad USB in this study. The suggested framework has been put to the test on a test bed. The recommended strategy is more secure than previous documented approaches in the literature, as shown by a comparison between the proposed scheme and the current methods. To obtain crucial data from the victim, an MITM attack may combine DNS poisoning and HTTPS hijacking.

Utilizing a VPN is one of the popular methods for securing network communication. VPN, however, is not the suggested option in this study due to its MITM and leakage concerns. However, Tor also has issues with serial leakage, which reduces its dependability. As a result, Shadow socks and its variants offer superior performance and security. Obfuscating algorithms can add an additional layer of security before an attacker obtains plaintext, even if an MITM attack compromises the encryption of Shadow socks traffic.

.

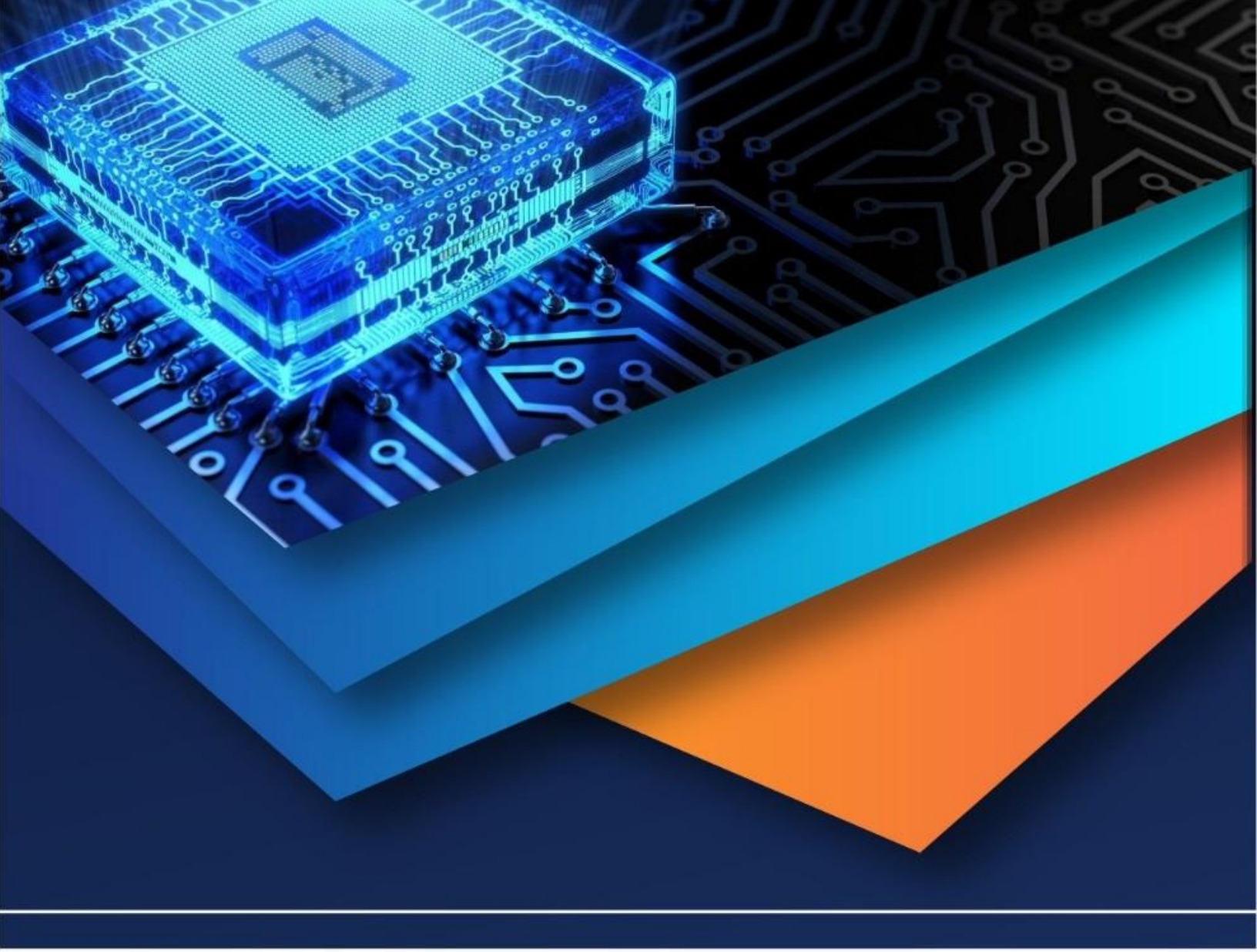
X. FUTURE SCOPE

In order to carry out various attacks, a bad USB can serve as a "Man in the Middle." To obtain crucial data from the victim, an MITM attack may combine DNS poisoning and HTTPS hijacking. Utilizing a VPN is one of the popular methods for securing network communication. VPN, however, is not the suggested option in this study due to its MITM and leakage concerns. However, Tor also has issues with serial leakage, which reduces its dependability. As a result, Shadow socks and its variants offer superior performance and security. Even if an MITM attack breaks the traffic encryption used by Shadow socks, the obfuscating algorithm can add an extra degree of security before the attacker gets access to plaintext.

REFERENCES

[1] Stephanie Blanchet “Bad USB, the threat hidden in ordinary objects”, 2018 Research.
[2] Seo, Jun-Ho “Analysis and Countermeasure for BadUSB Vulnerability”, 2017 IEMEK Journal of Embedded Systems and Applications.
[3] Pedro Brandao, Rohan Scanavez “Bad USB: why must we discuss this threat in companies?”, 2021 Research Review.
[4] Yeunsu Lee, Hyeji Lee, Kyungroul Lee & Kangbin Yim “Cognitive Countermeasures against BAD USB”, 2017 Part of the Lecture Notes on Data Engineering and Communications Technologies book series.
[5] Moon, Jong-Sub “Analysis and Countermeasure for BadUSB Vulnerability”, 2017 IEMEK Journal of Embedded Systems and Applications.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)