JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Wireless Bad USB Using Raspberry Pi Pico W

Reddyvari Venkateswara Reddy[1], Karlapalem Sujitha[2], Chilipi Chetti Balaji Sai[3], Pitla Sai Kumar[4], Rekhawar Sathvik[5]

[1] Associate Professor, Department of CSE (Cyber Security CMR College of Engineering & Technology, Hyderabad, Telangana, India
[2] Assistant Professor, Department of CSE (Cyber Security), CMR College of Engineering & Technology, Hyderabad, Telangana, India
[3, 4,5] Student, Department of CSE (Cyber Security), CMR College of Engineering & Technology, Hyderabad, Telangana, India

ABSTRACT: ***In this project, we explore the development of a Wireless Bad USB attack framework with the Raspberry Pi Pico, emphasizing how effective it can be as a tool for ethical hackers and security experts. Through the development of bespoke firmware for the Raspberry Pi Pico, we can discreetly retain wireless connectivity to a remote command and control server, while also enabling the device to smoothly mimic a USB HID and execute established malicious payloads. By bridging the gap between conventional USB-based exploits and remote exploitation, this novel method successfully gets around the restriction of physical access requirements. A secure wireless communication protocol, a flexible payload execution architecture, and an intuitive attack configuration and launcher interface comprise every part of our project. Furthermore, we will look into the idea of using state-of-the-art techniques for encryption to safeguard the communication link between the Raspberry Pi Pico assuring the privacy of crucial information and payloads. To further increase the effectiveness of the attack framework, we will investigate techniques to obscure the malicious payloads so that antivirus and intrusion detection software cannot recognize them.***

KEYWORDS- Cybersecurity, Ethical Hacking, Hashing, HID, Ducky Script, Raspberry Pi Pico W, Bad USB, Circuit Python, Payload, Micro Python.

1. INTRODUCTION

In wireless technology and cybersecurity, the Wireless Evil USB utilizing Pico W is a fresh and fascinating concept. With this concept, the Raspberry Pi Pico W, a compact, reasonably priced controller board with wireless connectivity, can be used to create a "Bad USB" device. A malicious USB is a kind of USB-based hack wherein, without the user's knowledge or consent, the operating system of a USB gadget is altered to execute dangerous instructions. In this scenario, the Pi Pico W is programmed to emulate a keyboard or other Human Interface Device (HID) so that it can execute stated commands or payloads when the gadget is connected to a victim machine. Furthermore, because the Wireless Bad USB using Micro-controller Pi Pico W does not require direct connection to the target system, it enables a discrete and covert manner of launching attacks. Because of this, it is very helpful in security assessments, penetration testing, and red teaming operations when confidentiality is imperative. The Raspberry Pi Pico W's wireless ability to carry off intricate sequences of actions from a distance is useful for security experts and ethical hackers to automate attacks. This improves the attack's strength and effectiveness because it reduces waiting time. In recent times, as One threat that's popping up is the Wireless Bad USB. It's a new kind of Bad USB attack. Attackers use it to find and use USB weak spots remotely. They don't even need to be near the device they're attacking. This 'remote' way of attacking is worrying because it happens over wireless ways of talking, like Bluetooth or Wi-Fi. These wireless attacks can be hard to find and stop. Our study aims to shine a spotlight on the possible dangers of Wireless Bad USB attacks on IoT gadgets. Networks splitting up, teaching users, secure startup, and app scans, are examples of strategies we're using to minimize such risks. With the Raspberry Pi Pico W's potential, we aim to help develop solid security steps and explore the idea of a Wireless Bad USB. This approach will help stop IoT devices from getting remotely exploited.

2. LITERATURE REVIEW

"Wireless Bad USB: A New Threat to IoT Security" (2020). Wilson discusses the characteristics, techniques for attack, potential defensive methods, and practical effects of Wireless Bad USBs. The study describes the need to know the risks associated with Wirelessly operated Bad USBs and implement suitable security measures to mitigate risks. He also speaks about how hazardous such devices could be shortly.

Threat Bad USB: A New Paradigm for Remote USB Exploitation" (2020), the author studies how remote USB exploitation capabilities of Wireless Bad USB. He explores the attack types and potential defect methods against the Wireless Bad USB,

bringing attention to the need for Companies to implement imperative security measures to secure opposite to these proliferating threats.

In an observation by M. Conti, M. Passarella, and F. Restuccia, titled "From Bad USB to Wireless Bad USB: An Analysis of the Security Threats and Potential Defenses" (2020), the Paper studies the evolution of Bad USB attacks to Wireless Bad USB. The paper speaks about the security threats by Wireless Bad USB. Then proposes potential defenses, such as secure boot and firmware validation mechanisms, network diversification and user awareness. It also analyzes the need for user education on the threats and mitigation methods.

BadUSB, the threat hidden in ordinary objects: In the paper, the author Stéphanie Blanchet talks about how dangerous bad USBs are. In the attacks using IoTs device firmware are manipulated to perform the attacks. More than 50% of microcontroller devices are vulnerable to such attacks. Yet, this flaw does not only affect secondary drives; it can infect any USB-equipped machine. Once the firmware gets modified, the malicious device can mimic any other device (e.g. keyboard, external hard disk, etc.) and take control of the computer, install a virus that could propagate to other USB peripheral devices, exfiltrate data, and spy on the user.

3. OBJECTIVE

In the bigger picture of Wireless Bad USB, our objective of using a Raspberry Pi Pico W is to show how vulnerable USB-connected devices are to remote exploitation based on wireless connections. With the use of this device's wireless connectivity, it is possible to get above traditional physical security measures by pretending to be a USB gadget device and remotely inserting malicious payloads into a victim machine. The achievement can be achieved through using the available various payloads. This also includes using scripts known as ducky scripts. To take benefit of exploiting the vulnerabilities in the protocol, the USB drivers, or the firmware/software of the target device, the Raspberry Pi Pico W can be set up to function as a USB device, such as a Mouse or secondary storing device. The vulnerability in a USB machine that enables it to assess wireless remote access can be demonstrated using the Raspberry Pi Pico W with WLan Bad USB. By impersonating a USB machine and leveraging the equipped device's built-in WiFi, it's possible for remote injection of malicious software into a target or victim device, defeating traditional port security measures.

4. PROBLEM DEFINITION

The Wireless Bad USB utilizing Raspberry Pi Pico W project seeks to solve two issues: Deficiency of Automation:Automating repetitive operations on an electronic system is essential in many situations. Standard methods might be ineffective and time-consuming. An efficient USB gadget can increase efficiency by replicating a mouse and performing programmed orders, or "payloads," at superhuman speeds. Security on the Internet Failures: Computers have an intrinsic trust for USB devices. A threatful USB device may take advantage of this trust by sending payloads or viruses that can unintentionally alter the system. Also, the tech would let you make a dreadful device. In short, to overcome such issues, our project uses Raspberry Pi Pico W, a tool that can both expose possible weaknesses in security and automate operations. The tool can now be operated remotely due to its wifi capabilities, which adds even more capability and complexity.

5. METHODOLOGY

5.1 Installation of Circuit Python

Here, first we have to install Circuit Python in the device.

5.2 Writing the Duckyscript

It can be used to write the executable code.

5.3 Transferring the Payload

Create a root account in the device, and transfer all the payloads.

5.4 Setting up AP

set up the AP and give a password to it.

5.5 Connect it the Attacker's

With the help of AP you can execute remote code.

6. IMPLEMENTATION

In this project architecture had three many types of payloads.It can vary according to the need of the operation to be performed.

It has the capability of holding multiple payloads at a time.

6.1 Connecting to Victim machine

With the help of a USB cable we can attach our Raspberry pi pico w into the victim machine's port.

6.2 Accessing the Device

With the AP of raspberry pi pico w,we will be able to connect the device..

6.3 Selection of Payload

Once you have access to AP.You will be able to access all the payloads present in the gadget.Once you select the payload it will start running in the victim's machine.

6.4 Exploitation

This is the final stage,wherein you will be able to control the victim's machine.It lets to access the data from the victim

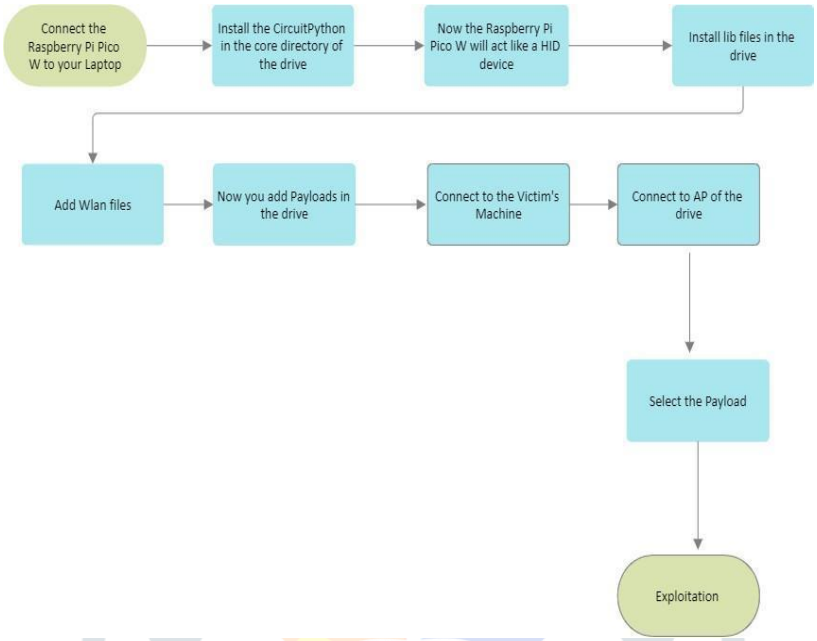

Fig.1 Workflow

7. RESULT

Once the attack is successful you can have access to the victim's display and control it remotely. You will be able to see it on your machine.

```
Example:
        BEGINNING OF PAYLOAD
        ... Payload Documentation...

        REM CONFIGURATION
        REM REQUIRED - Provide URL used for Example
        DEFINE #MY_TARGET_URL example.com

        REM OPTIONAL - How long until payload starts; default 5s
        DEFINE #BOOT_DELAY 5000

        DELAY #BOOT_DELAY
        ...
        STRING #MY_TARGET_URL
        ...
```

Fig.2 Duckyscript

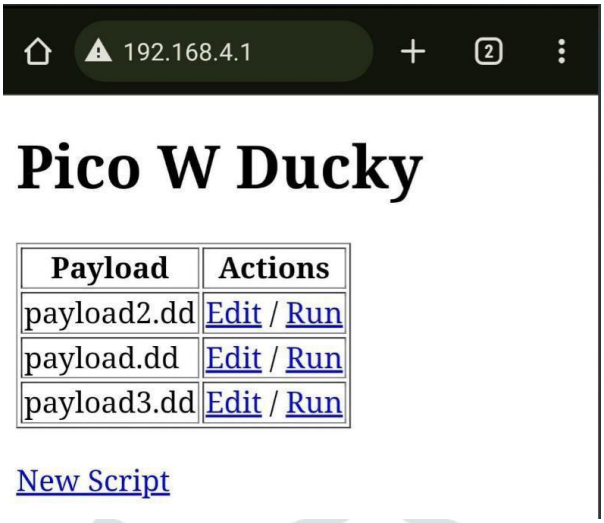

Fig.3 Payloads

8. CONCLUSION

Our project "Wireless Bad USB Pi Pico W" aims to spread awareness about how dangerous open ports are. It also involves various payloads and structures which are essentially useful for penetration testing. Lastly, we want to state that our intention for the creation of this projection is for research and development only. We don't appreciate the wrong usage. We have lots of payloads in our project. These are paired with vital structural elements for testing. They're used to gauge the system's strength. We've designed the payloads to mimic potential threats. They work like a security checkup or practice drill. So, the system experts find and fix weak areas before any real problems. It's like we provide a spotlight for them to scrutinize their system and expose vulnerabilities. That way, security concerns shrink while their protective shields grow stronger.

REFERENCES

[1] Bad USB, the threat hidden in ordinary objects, June 2018 by Stéphanie Blanchet.

[2] MakingWhitelistingBased DefenseWorkAgainst BadUSB:https://dl.acm.org/doi/10.1145/3289100.328912

[3] Pico-Ducky: https://github.com/dbisu/pico-ducky

[4] https://www.linkedin.com/pulse/repost-turning-usbperperipherals-badUSB-desmond-Israel

[5] https://github.com/hak5darren/USBRubberDucky/wiki/PayloadWIN10-Disable-WindowsDefender

[6] https://micropython.org/download/rp2-pico-w/

[7] https://gainsec.com/2020/04/27/generating-a-msf-reverseshell-qualities-9/

[8] https://github.com/davidbombal/hak5/blob/

[9] View of Bad-USB Why must we discuss such malicious threats in Computer

[10] USB Captcha In Preventing (un-)conventional attacks from used USB devices in industrial systems