**Bluetooth Based HID Device**

# Bluetooth Rubber Ducky

**By, Mayank Parasramka, Ladva Umanshiva Hiteshiva, Yasir Usmani**

## Introduction

This project demonstrates how to turn a Raspberry Pi Zero W into a Bluetooth Human Interface Device (HID) that can interact with an Android device. Specifically, the device will use a custom payload to open and play a YouTube video on an Android device. We will use the `pybluez` library and the `BlueDucky.py` script to achieve this.

**What is a Human Interface Device (HID)?**

A Human Interface Device (HID) is a type of computer device that interacts directly with humans and translates human actions into electronic signals that a computer or other digital system can understand. Common examples of HIDs include keyboards, mice, game controllers, and touchscreens.

HID devices use the HID protocol, a standardized protocol defined by the USB Implementers Forum (USB-IF), which allows for the easy integration of peripheral devices with a host computer or system. The HID protocol is designed to be simple and extensible, making it easy to develop new types of input devices without needing to create new drivers for each device.

**Key Characteristics of HID Devices:**

1. **Standardized Protocol**: HID devices follow a standardized protocol, which ensures compatibility across different systems and platforms. This means that a USB keyboard, for example, will work with any computer that supports the USB HID protocol without requiring special drivers.
2. **Ease of Use**: Because the HID protocol is standardized, operating systems come with built-in support for many types of HID devices. This plug-and-play capability makes it easy to add new input devices to a system.
3. **Device Reports**: HID devices communicate with the host system using "reports." These reports contain data about the actions performed by the user, such as key presses or mouse movements. The format of these reports is defined by the HID protocol and can vary depending on the type of device.
4. **Versatility**: The HID protocol is versatile and can be used for a wide range of input devices beyond just keyboards and mice. This includes game controllers, remote controls, biometric readers, and more.

**HID over Bluetooth:**

Bluetooth HID is a variant of the HID protocol designed for Bluetooth devices. It allows wireless HID devices to communicate with a host system over a Bluetooth connection. Bluetooth HID is commonly used for wireless keyboards, mice, and game controllers.

In this project, we leverage the Bluetooth HID functionality to turn the Raspberry Pi Zero W into a wireless HID device. By doing so, we can send keyboard input to an Android device, automating the process of opening a web browser, navigating to YouTube, and playing a specific video. This showcases the power and flexibility of the HID protocol and its applications in automation and control.

By using the `pybluez` library and the `BlueDucky.py` script, we can create custom payloads that simulate complex user interactions, making it possible to control the Android device programmatically. This setup opens up a wide range of possibilities for automating tasks and creating interactive systems using Bluetooth-enabled Raspberry Pi devices.

# Requirements:

- Raspberry Pi Zero W
- MicroSD card (8GB or larger) with Raspberry Pi OS installed
- Bluetooth-compatible Android device
- Power supply for the Raspberry Pi
- Internet connection for the Raspberry Pi (for installing dependencies)

# Hardware Setup:

**About Raspberry Pi Zero W:**

The Raspberry Pi Zero W is a compact and affordable single-board computer developed by the Raspberry Pi Foundation. It is part of the Raspberry Pi family, known for providing powerful computing capabilities in a small form factor. The "W" in Raspberry Pi Zero W stands for wireless, indicating its built-in Wi-Fi and Bluetooth capabilities. Despite its small size, the Raspberry Pi Zero W is a versatile device suitable for a wide range of applications, from simple DIY projects to more complex embedded systems.

**Key Features**

1. **Processor:**
    - **CPU:** Broadcom BCM2835
    - **Architecture:** ARM1176JZF-S
    - **Clock Speed:** 1 GHz
2. **Memory:**
    - **RAM:** 512 MB LPDDR2 SDRAM
3. **Networking:**
    - **Wi-Fi:** 802.11 b/g/n
    - **Bluetooth:** Bluetooth 4.1, Bluetooth Low Energy (BLE)
4. **Ports and Connectivity:**
    - **USB:** 1 micro-USB port for data and power, 1 micro-USB OTG port
    - **HDMI:** Mini HDMI port for video output
    - **GPIO:** 40-pin GPIO header (unpopulated)
    - **Camera:** CSI camera connector

    - **Storage:** MicroSD card slot for storage and operating system

5. **Dimensions:**

    - **Size:** 65mm x 30mm x 5mm
    - **Weight:** Approximately 9 grams

**Advantages**

- **Affordability:** One of the most cost-effective computers available.
- **Compact Size:** Small and lightweight, making it easy to embed in various projects.
- **Wireless Capabilities:** Built-in Wi-Fi and Bluetooth enable wireless communication without additional peripherals.
- **Versatility:** Suitable for a wide range of applications, from simple DIY projects to complex embedded systems.
- **Community Support:** Extensive community support and a wealth of resources available for learning and troubleshooting.

**Limitations**

- **Limited Processing Power:** Less powerful than other Raspberry Pi models, making it less suitable for resource-intensive applications. By less it is genuinely very less, like firefox fox took 15 minutes to just show up and it couldn't handle a google search, so it is best to use it only to execute scripts.
- **USB Port:** Limited connectivity options without USB Type-A hubs.
- **No Built-in Storage:** Requires a microSD card for storage and the operating system.

Overall, the Raspberry Pi Zero W is a powerful and flexible tool that provides significant functionality in a tiny and affordable package, making it a popular choice for hobbyists, educators, and developers alike.

**Setup Guidlines:**

The following steps were to be followed for an head-less setup of raspberry pi zero W

- Prepare the Raspberry Pi Zero W:
- Install a Raspberry Pi Imager and use it to install the OS to the SD card.
- Insert the microSD card with Raspberry Pi OS(32 bit) into the Raspberry Pi.
- Connect a micro-USB power supply to the Raspberry Pi.
- Ensure the Raspberry Pi has internet access through Wi-Fi. Ensure that you connect to 2.4 GHz network and not 5Ghz network as it is not compatible with the Raspberry Pi.
- Enable SSH (optional but recommended): Create an empty file named ssh (no file extension) in the root directory of the boot partition of the SD card. This enables SSH access. Latest generation imager have inbuilt functionality to enable ssh.

The above method is known as the headless method for setting up the raspberry pi. Although it works in most of the cases, it didn't for us so we chose to go with direct way of setting it up using the mini HDMI cable and an external monitor. The process goes as follows,

- After installing the OS in the SD card using the imager, we connect the raspberry pi to an external monitor via mini-HDMI to HDMI cable, and boot it on by powering it from the power port (micro-USB).

- Next we connect the second micro-USB (used for data transfer and power) to an wireless mouse and keyboard using 2.4G dongle.
- After booting it on we setup the wifi and bluetooth, and update/upgrade the terminal using sudo apt update & sudo apt upgrade.

## Software Setup:

The software setup for this project involves installing and configuring various tools and libraries on the Raspberry Pi Zero W. This setup transforms the Raspberry Pi into a Bluetooth Human Interface Device (HID), which can simulate keyboard and mouse inputs to interact with an Android device. The core components of the software setup include updating the Raspberry Pi OS, installing necessary dependencies, configuring Bluetooth settings, and preparing the script (BlueDucky.py) to execute our custom payload. This setup is essential for the Raspberry Pi to communicate with the Android device over Bluetooth and send the desired keystrokes to automate tasks like playing a YouTube video.

Firstly update the system:

```
sudo apt update
sudo apt upgrade -y
```

Keeping the Raspberry Pi's software up to date is crucial for security and compatibility reasons. This command updates the package list and upgrades all installed packages to their latest versions.

The most important library for this project is `Pybluez`.

**About `Pybluez`:**

`Pybluez` is a Python library that provides easy access to Bluetooth operations. It enables developers to perform various Bluetooth tasks such as device discovery, service browsing, and data exchange. In this project, `Pybluez` is crucial for establishing Bluetooth communication between the Raspberry Pi and the Android device. By using `Pybluez`, we can create custom Bluetooth applications, including our HID device, with minimal effort.

`Pybluez` abstracts the complexities of Bluetooth programming, offering a simple API to interact with Bluetooth devices. This makes it an essential tool for developing Bluetooth-enabled applications in Python, enhancing the Raspberry Pi's capabilities to function as an HID device that can execute our custom payload.

**Installing `Pybluez`:** Run the following commands in the terminal of debian-like (Linux based) Raspberrypi OS (32-bit) inorder to install the library:

```
sudo apt install python3-pip bluetooth bluez libbluetooth-dev -y
pip3 install pybluez
```

alternatively,

```
sudo git clone https://github.com/pybluez/pybluez.git
```

now go int to pybluez folder,

```
cd pybluez/
```

verify the bluetooth fuctionality and connectivity using the following commands:

```
sudo hciconfig hci0 up
sudo hcitool scan
```

Installing all the other relevant scripts and tools using the follwing,

```
git clone https://github.com/bluez/bluez.git
```

## Writing a payload in Ducky Script:

**What is a Ducky Script?** A Ducky Script is a simple scripting language used to automate keyboard input. It is commonly used with devices like the USB Rubber Ducky, a keystroke injection tool that can execute pre-defined scripts on a target computer. Ducky Scripts are composed of a series of commands that simulate keystrokes, mouse movements, and other input events, allowing the user to automate tasks such as opening applications, typing text, and executing commands.

***What is a Payload?*** A payload, in the context of Ducky Script, refers to the specific set of commands or script that is executed by the device. It is the actual code that performs the desired actions on the target machine. For example, a payload can be a script that opens a web browser and navigates to a specific website. In this project, the payload will be a script that opens the browser on an Android device and navigates to a YouTube video.

Ducky Script to open a youtube video on Android:

```
REM Opens a private browser to youtube.com
DELAY 200
ESCAPE
GUI d
ALT ESCAPE
GUI b
DELAY 700
REM PRIVATE_BROWSER is equal to CTRL + SHIFT + N
PRIVATE_BROWSER
DELAY 700
CTRL l
DELAY 300
STRING https://www.youtube.com/watch?v=dQw4w9WgXcQ
DELAY 300
ENTER
DELAY 300
```

### Pairing the android device to Raspberry Pi:

Before running the script the Raspberry Pi must be paired to the device, following the steps below:

- Type `bluetoothctl` and press Enter to open Bluetooth control
- At the `[bluetooth]#` prompt enter the following commands:

```
discoverable on
pairable on
agent on
default-agent
scan on
```

- Wait for a message to appear showing the Android phone has been found:

```
[NEW] Device 12:23:34:45:56:67 devicename
```

- Type pair with the mac address of your Android phone:

```
pair 12:23:34:45:56:67
```

- On your Android phone and Raspberry Pi.
- Confirm the passcode. Your device is now paired.

### Running the script:

Go to directly where the BlueDucky.py script is present and run it with root access so that it can access the bluetooth freely:

```
sudo chmod 755 BlueDucky.py
sudo python3 BlueDucky.py
```

Now the BlueDucky environment opens up and choose the host android device (which was paired before) and run the created payload on it. This successfully runs and the host android phone started running the youtube video.

## Extra Informations:

This script is based on an recently discovered vulnerbality serialized as **CVE-2023-45866** in the vulnerability database. ***CVE-2023-45866*** is a security vulnerability identified in a specific software or system component. The Common Vulnerabilities and Exposures (CVE) system provides a reference method for publicly known information-security vulnerabilities and exposures. ***CVE-2023-45866*** follows this naming convention, where

"CVE" stands for Common Vulnerabilities and Exposures, "2023" indicates the year the vulnerability was disclosed, and "45866" is the unique identifier for this specific vulnerability.

**Summary:** CVE-2023-45866 describes a security flaw that could potentially allow an attacker to compromise the affected system. Details about the exact nature of the vulnerability, including the affected software, the type of flaw, and the potential impact, are typically provided by the organization that discovered or reported the vulnerability. CVE-2023-45866 represents a specific security vulnerability that needs to be addressed to ensure the continued security and integrity of the affected systems. Staying informed about such vulnerabilities and promptly applying necessary updates or mitigations is crucial in maintaining robust cybersecurity defenses. For detailed information, users should refer to the advisories and bulletins issued by the software vendors or security organizations.

**Potential Impact:** The impact of CVE-2023-45866 can vary based on several factors, including the specific nature of the vulnerability, the affected system, and the environment in which the system is deployed. Common impacts of such vulnerabilities include unauthorized access to sensitive data, system crashes, execution of arbitrary code, privilege escalation, and other forms of security breaches.

**Mitigation and Remediation:** To mitigate and remediate CVE-2023-45866, it is essential to:

- Apply any available patches or updates provided by the software vendor.
- Implement security best practices, such as restricting access, using firewalls, and monitoring system activity.
- Follow any specific recommendations or guidelines issued by the organization that disclosed the vulnerability.

## Major Issues/Fixes:

During the project numerous issues and bugs were encountered and here are the major ones:

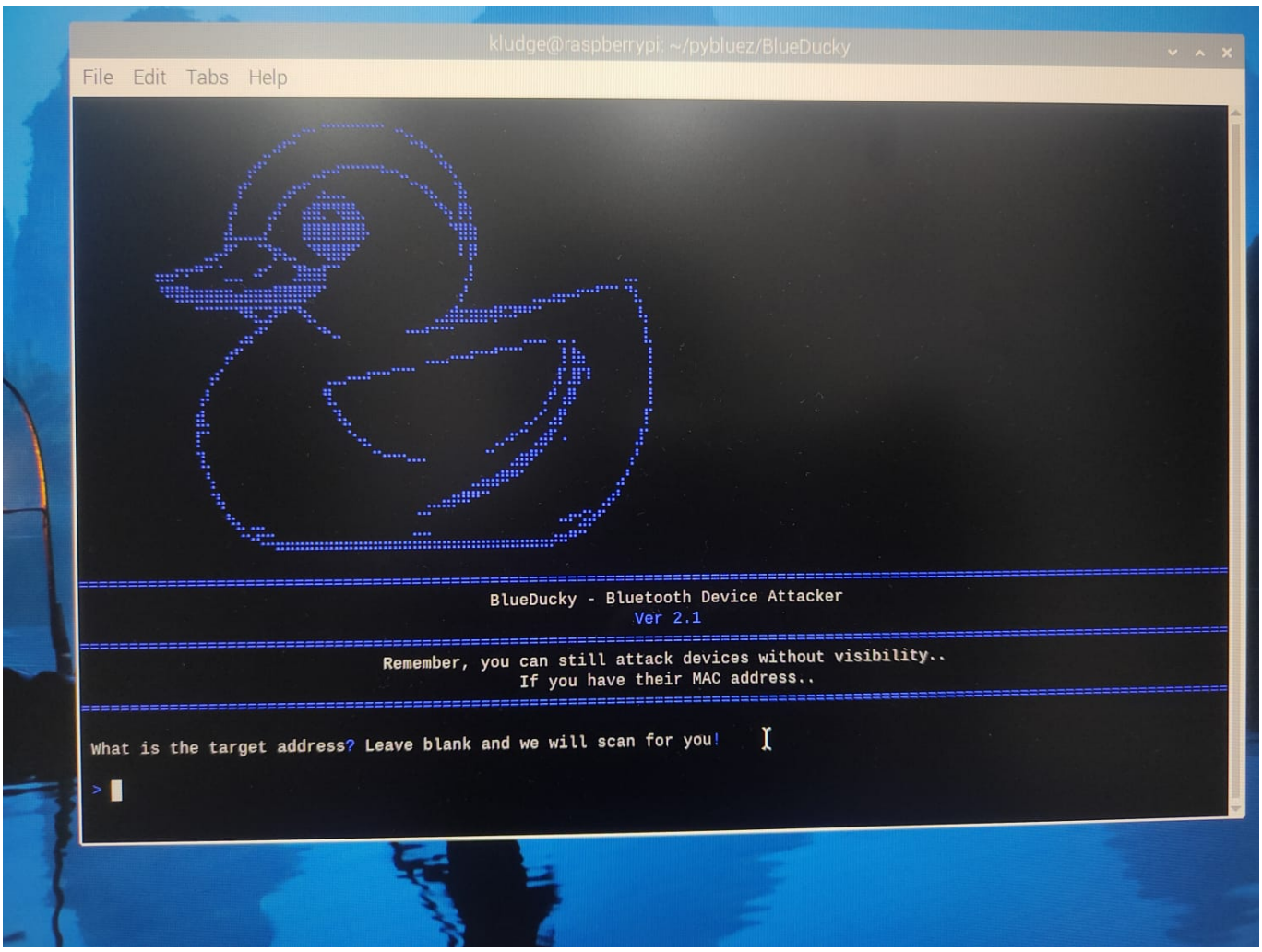
- Setting up the Raspberry Pi using headless setup method was the most difficult step, despite countless attempts the raspberry pi was unable to connect to IITH wifi network or even mobile hotspot, this issue was rectified using manual setup method using mini-HDMI cable and an external monitor.
- Bluetooth connectivity with Raspberry Pi Zero W was another major issue that was rectified by updating the system and installing the necessary bluetooth drivers on the OS.
- Lastly, getting the code to be able to use bluetooth flawlessly was the biggest issue, and it was rectified by giving necessary root accesses and making necessary changes in the code with appropriate model of the bluetooth device ("HCI0" or "HCI1").

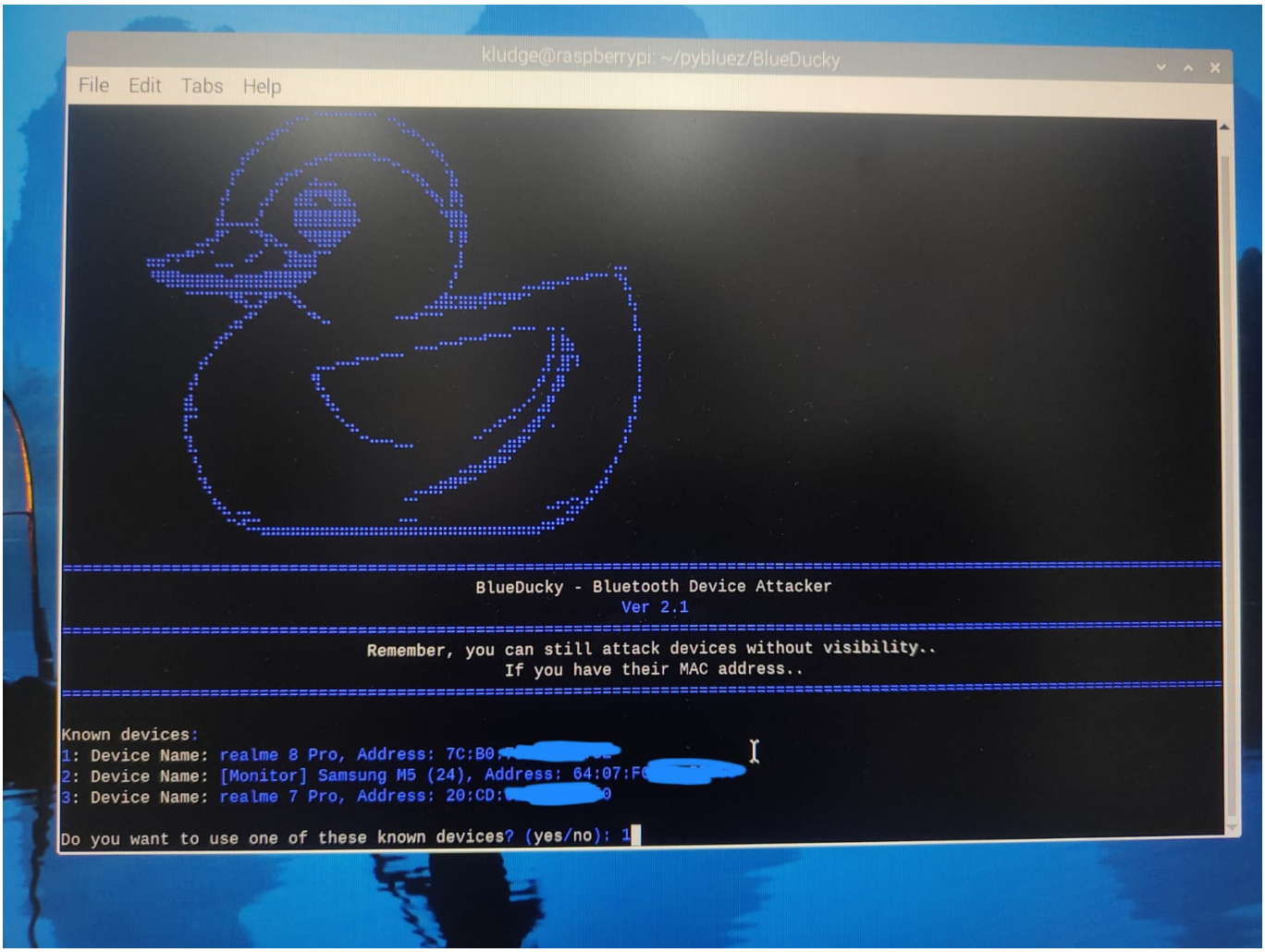
## Final Conclution:

We have successfully created a Bluetooth-based HID device using a Raspberry Pi Zero W that can run a custom payload to play a YouTube video on an Android device. This project demonstrates the power and flexibility of Raspberry Pi in combination with Python libraries to create custom HID devices.

We can further modify the payload to perform different tasks or to control other applications on the Android device, extend it further to windows. Moreover this can be used in Remote control via web interface and in gaming as well. The possibilities are endless!!


Welcome to the
Raspberry Pi Desktop
Powered by Raspberry Pi OS (32-bit)
Release 5.2 - March 2024


kludge@raspberrypi: ~/pybluez/BlueDucky
File Edit Tabs Help
BlueDucky - Bluetooth Device Attacker
Ver 2.1
Remember, you can still attack devices without visibility..
If you have their MAC address..
What is the target address? Leave blank and we will scan for you!
>


kludge@raspberrypi: ~/pybluez/BlueDucky
File Edit Tabs Help
BlueDucky - Bluetooth Device Attacker
Ver 2.1
Remember, you can still attack devices without visibility..
If you have their MAC address..
Known devices:
1: Device Name: realme 8 Pro, Address: 7C:B0
2: Device Name: [Monitor] Samsung M5 (24), Address: 64:07:F
3: Device Name: realme 7 Pro, Address: 20:CD:
Do you want to use one of these known devices? (yes/no): 1