Bad USB Attacks

Calvin Schmeichel
St. Cloud State University
Cybersecurity
Calvin.schmeichel@go.stcloudstate.edu

Will Novotny
St. Cloud State University
Cybersecurity
Will.Novotny@go.stcloudstate.edu

I. Abstract

The widespread use of USB devices in everyday life has made them a primary target for cybercriminals. One of the most significant threats is the emergence of BadUSBs, which are USB devices that have been modified to perform malicious activities. BadUSBs can infect systems with malware, steal sensitive data, and launch attacks against critical infrastructure. This research aims to investigate the current state of BadUSB attacks, their impact on different sectors, and explore novel approaches to detect, prevent, and mitigate BadUSB attacks, with a focus on improving network resilience, reducing downtime, and enhancing cybersecurity strategies. We will then apply our findings into a lab experiment that will allow us to apply the topics we have been researching into a simulated real world environment.

II. Problem Domain

A BadUSB attack is a type of cyber attack where a USB device is manipulated to act as a keyboard or mouse, allowing an attacker to execute malicious commands on a target computer. This type of attack can be used to install malware, steal sensitive data, or even take control of the target computer. BadUSB attacks are a growing threat to cybersecurity, and many organizations are looking for ways to protect against them.

The main problem that we identified that was most affected by a BadUSB attack was the user domain and the workstation domain.

User Domain: The user domain refers to the individuals or groups who use a computer system and the data it contains. It encompasses the human factors that can affect the security of a system, including user behavior, training, and awareness. The user domain is a critical aspect of cybersecurity, as human error or malicious actions can compromise the confidentiality, integrity, and availability of data. Effective security measures in the user domain include policies, training, and access controls to ensure that only authorized users can access sensitive information.

A user domain can be affected by a BadUSB by, a BadUSB device can mimic a keyboard, it can execute keystrokes and commands on the user's computer without their knowledge or consent. This means that sensitive information such as login credentials, financial data, and personal information can be stolen by the attacker. Additionally, a BadUSB device can be programmed to install malware or ransomware on the user's computer, which can result in data loss or extortion.

Workstation Domain: The workstation domain is a part of a computer network that includes all endpoints, such as desktops, laptops, and mobile devices that are used by the employees or other authorized personnel. The domain is responsible for managing the software and hardware components of these endpoints, ensuring their proper configuration, and implementing security measures to prevent unauthorized access or data breaches. The domain also includes policies and procedures for the management of workstations, such as regular updates and patching, backup and recovery plans, and monitoring for security incidents.

Effective management of the workstation domain is essential for ensuring the security and stability of the entire network infrastructure.

A workstation domain can be affected by a BadUSB by, once a BadUSB device is connected to a workstation, it can also spread malware to other connected devices or network shares, making it difficult to contain the attack. The attack can also exploit vulnerabilities in the operating system or software applications, leading to system crashes or data corruption.

III. Problem Statement

> "BadUSB devices are a growing threat to computer security, and their impact can be devastating. Existing research has mostly focused on individual attacks and specific mitigation strategies, but there is a need for a comprehensive and systematic approach to understanding the security implications of BADUSB devices. Therefore, there is a pressing need for research that investigates the security implications of BADUSB devices and develops effective detection and mitigation techniques."

IV. Research Questions

a. What are USBs

USB stands for Universal Serial Bus, which is a type of interface that allows data transfer between electronic devices. A USB port can be found on a wide range of electronic devices, such as computers, smartphones, printers, and game consoles.

A USB flash drive, also known as a USB stick, thumb drive, or memory stick, is a small portable storage device that uses a USB port to connect to a computer or other electronic device. It typically has a capacity ranging from a few gigabytes to several terabytes and can be used to store and transfer files and data.

USB devices can also include other peripherals, such as keyboards, mice, webcams, and audio devices, which can connect to a computer's USB port for use. USB devices are generally easy to use and convenient, but they can also be a potential security risk, especially when they are used to introduce malware or execute unauthorized commands, as in the case of a BadUSB attack.

b. What is a BADUSB

A BadUSB is a type of attack that involves manipulating the firmware of a USB device, such as a flash drive or keyboard, to perform malicious actions on a computer or other electronic device. This attack can allow an attacker to take control of the targeted system, steal sensitive information, install malware or ransomware, or perform other malicious actions. Because BadUSB attacks can be difficult to detect, and the malicious code can be hidden within the USB device, they can be an effective tool for cybercriminals seeking to compromise a system or network.

The most common type of BadUSB is a Rubber Ducky, which we will be using in this project. According to, "Forensic Log-Based Detection for Keystroke

Injection BadUSB Attacks", "The most well-known BadUSB attack vector is probably the commercial "Rubber Ducky" which initially started as a sysadmin gadget to automate mundane tasks. This platform evolved into the most notorious attacker gadget with a series of community-backed payloads whose main capabilities include dual usage as a USB Stick, data exfiltration, HID Interaction (Keystroke Injection) and even its own scripting language, DuckyScript. This is backed by a USB2.0 hardware interface and support for USB-c. Some of the most advanced features include, copying payload to itself, "OUT endpoint" usage via "Lock Keys" "spamming", Keystroke Reflection and even features like VendorID and ProductID spoofing (Karantzas, Forensic log based detection for keystroke injection "Badusb" attacks 2023)." [11]

c. How do bad USBs defeat protections in place

BadUSBs defeat protections in place by pretending to be a USB drive or something else. That could be a storage device or a regular flash drive. According to the article, "Defending Against Malicious USB Firmware with GoodUSB", " In the BadUSB attack, a malicious USB device registers as multiple device types, allowing the device to take covert action on the host. For example, a USB flash drive could register itself as both a storage device and a keyboard, enabling the ability to inject malicious scripts (Defending against malicious USB firmware with goodusb)." [7]

The other way that a BadUSB defeats protections is in place, because USB device firmware cannot be scanned by the host, antivirus software cannot detect or defend against BadUSBs.

d. Who is most targeted

BadUSBs targets individuals, businesses, and government organizations. Organizations containing sensitive information are going to be the most targeted. Anyone can fall victim to a BadUSB attack, regardless of their profession or level of expertise. It's always a good idea to take precautions such as only using trusted USB devices, keeping software and security measures up to date, and being cautious when plugging in unfamiliar USB devices.

e. How to defend against them?

There are many ways to defend against BadUSBs. Some of the most popular ways to defend against them are USB port blocker, using special programs to monitor typing speed, and restricting access to elevated command prompt.

USB Port Blocker: "USB port blockers are an effective way to deter users from connecting unauthorized USB devices that may contain malicious payloads without their knowledge. In the case of a BadUSB attack, an attacker is less likely to target systems where USB port blockers are installed."

With a USB port blocker installed. they are most likely going to be on systems with sensitive files and come with a special key that unlocks and locks the device once installed (ManageEngine, BadUSB attack prevention).” [6] They are a great way to defend against a BadUSB attack.

Using special programs to monitor typing speed: With programs like DuckHunter it is designed to run in the background and look for typing speed. According to the article, What is BadUSB Attack and How to Prevent, “Since BadUSB devices type at speeds that are practically impossible for humans to type at, the program effectively blocks keyboard input when a BadUSB attack is detected (ManageEngine, BADUSB attack prevention).” [6] This is a great way to detect a BadUSB. However, using this method means that these programs sometimes take a few milliseconds to detect an attack.

Restricting access to elevated Command Prompt: Running Command Prompt as an administrator unlocks a lot of access to a set of actions that can be performed on a computer. Setting a password for using elevated Command Prompt stops the BadUSB programmed to seek administrative privileges. With the password in place BadUSBs cannot access any sensitive information. It is a great way to keep your systems safe.

f. How to write a BADUSB script?

There are many ways to write BAD USB. Such as Bash, Python and Powershell. But the most commonly used scripting language is DuckyScript from Hak5. Since the Rubber Ducky is the most common BAD USB device. The Flipper Zero also uses a fork of the DuckyScript language too. So DuckyScript will be our focus for this paper and the lab.

Here is a list of basic commands:

```
STRING : “Enters Keystroke injects (types) a
series of keystrokes. keystroke injects (types) a
series of keystrokes.”

Modifier keysModifier keys: “Can be sent in
combination with another key to perform a special
function. combination with another key to perform
a special function.” Example: “ctrl + v” for
paste.
```

- SHIFT
- CTRL
- ALT
- GUI (Windows Key)

```
DELAY: instructs the USB Rubber Ducky to
momentarily pause execution of the payload.
instructs the USB Rubber Ducky to momentarily
```

```
pause execution of the payload. This can be
useful when you need to wait for an application
to load on a slower computer.
```

These are just a few commands and functions. The DuckyScript is simple but at the same time quite powerful in the right hands.

g. What is the goal/data?

Some people who use BadUSBs are penetration testers, system administrators, and other cybersecurity professionals to get work done faster. Other people who use them can be cybersecurity criminals who want to gain access to a computer and steal information from it.

According to the article, What is BadUSB Attack and How to Prevent, "BadUSBs are capable of typing out thousands of characters in an instant and quickly runs through prompts, which is why penetration testers often use these devices to gain control over target computers without having to manually type out lengthy scripts. Even system administrators often resort to BadUSB devices while setting up new computers. Why spend hours on end going through prompts when they can have a BadUSB do it for them?

Apart from the legitimate uses of BadUSB, its potential of being weaponized when in the wrong hands is still a largely debated topic in the cyber community today. Cyber criminals often use BadUSB as an external tool to inject malicious scripts that are designed to seek administrative privileges, steal passwords, or download malware to a computer (ManageEngine, BadUSB attack prevention)." [6]

From this we can see that cyber criminals want to gain access to the administrator privileges to steal passwords and install malware on the computer. There is a high chance that hackers may want to steal information like personal information (SSN, bank information, addresses, emails, and more).

h. What is a Flipper Zero?

A Flipper Zero is a "swiss army knife" penetration testing tool. It has many hardware components such as NFC, Bluetooth, Sub-Ghz, RFID, BAD USB (Which we will focus on) and more.

This fun little tomodachi looking toy is fun outside but possibly dangerous inside. It is a great tool for pen-testers to play with.

We will be using its BAD USB tools during the lab section of the report to show off some scripts.

i. What is a Rubber Ducky?

A Rubber Ducky is a custom USB drive created by Hak5 that has extra tools for BAD USB scripts such as OS detection, keyboard cloning, normal USB mode and more. This harmless looking USB drive can do a ton of damage in the

wrong hands. But it is an awesome tool for cyber security students to learn from. We will be using the tools during the lab section of the report to show off some scripts.

V. Objectives and Scope

Objective: The objective of this project is to design and implement a cybersecurity solution to protect against Bad USB attacks. The solution should be easy to use and effective in preventing unauthorized access to a target computer.

We will focus on USB drives as a whole. We will talk about how they can be used in a malicious way and how easy they can be used to implement malware on a computer's system/network. We will go over and implement a Rubber Ducky USB and create a script with a lab that will showcase what a Rubber Ducky USB drive can do.

A rubber ducky USB is a malicious device used in cybersecurity projects to execute keyboard commands on a target computer. It appears as a normal USB drive but acts like a keyboard, bypassing some security measures. It can execute predefined keystrokes to perform actions like installing malware or stealing information. It's often used in penetration testing to assess the effectiveness of security measures. Its small size and ability to execute automated commands make it a potent tool for hackers.

We will show how to use a BadUSB scripting language in our lab and how it works with a video showing what happens when you plug that USB in a laptop or desktop. It will show how the computer will automatically start reading the script that we create and demonstrate what happens to that computer.

a. Scope: The scope of this project will include the following:
b. Research: Conduct thorough research on BadUSB attacks, including how they work, the types of attacks that can be carried out, and how they can be prevented.
c. Design: Based on the research, design a BadUSB device that can be used to execute an automated attack on a target computer.
d. Develop: Develop the BadUSB device using a microcontroller such as an Rubber Ducky, and program it to execute the automated attack.
e. Implement: Implement the BadUSB attack in a controlled environment using a target computer and the developed device.
f. Test: Conduct a series of tests to determine the effectiveness of the BadUSB attack and identify any weaknesses in cybersecurity measures.
g. Analyze: Analyze the results of the testing and identify areas for improvement in cybersecurity measures.
h. Mitigate: Develop and implement strategies to mitigate the vulnerabilities identified during the testing.

VI. Research Contribution

Our research contribution to the field will be us gaining a better understanding of the attack. Since our wide and thorough research will provide so much information we will aggregate all of it into a short and concise research paper that anyone can read at a high level.

We would also like to provide simple and easy to follow lab instructions during our experiment section to help future research easily set up a home lab and be able to learn in a real environment on how a BAD USB attack works. This will be free and hopefully provide a great point of research for future generations.

VII. Expected results

We expect to learn more about the USB protocol and scripting in general. But I would like to learn a few more things.

- **IEEE Research Question 1: "**How damaging can a Bad USB attack be?"
    - Since we have done research on other attacks before we would like to see the comparable damage a BAD USB can do compared to other forms of intrusion.

- **IEEE Research Question 2:** "How effective are BAD USB Attacks"
    - We would like to find research papers that have tested the validity of BAD USB's in an enterprise environment. This way we can gather more information from the real world.

VIII. Literature Review

A. "5-4 Studies on Countermeasures for Malicious USB Devices" [12]
B. "Create Your Own Bad USB" [13]
C. "Software Firewall to Protect against (Bad)USB Devices" [14]
D. "Securely Transferring Data from BadUSB Devices" [15]

IX. Methodology

Our main research design for this project was to have a mixed method of quantitative, qualitative approaches through occurring broad information on the topic. We then wanted to apply these findings to our lab and have an equally quantitative and qualitative look at those.

The sampling we used for this study was referencing mainly research documents on the subject of what was written from 2019 or newer.

The data we collected for this study was from the data sets used in the research documents mentioned before. Most of them contained many graphs, spreadsheets and math formulas which were useful to us. Using this data, we can then analyze it to better answer our research questions mentioned above.

One of our limitations while conducting this research was time. When starting this project, we were quite new to the topic, but as we drove deeper into its wealth of knowledge we then realized how much there is to learn. Another reason we tried to keep the information we did gather high level.

The Validity and Reliability of our research was of utmost importance to us. We only used resources provided to us through the St. Cloud State University's online library, Google scholar and other students' research conducted on campus.

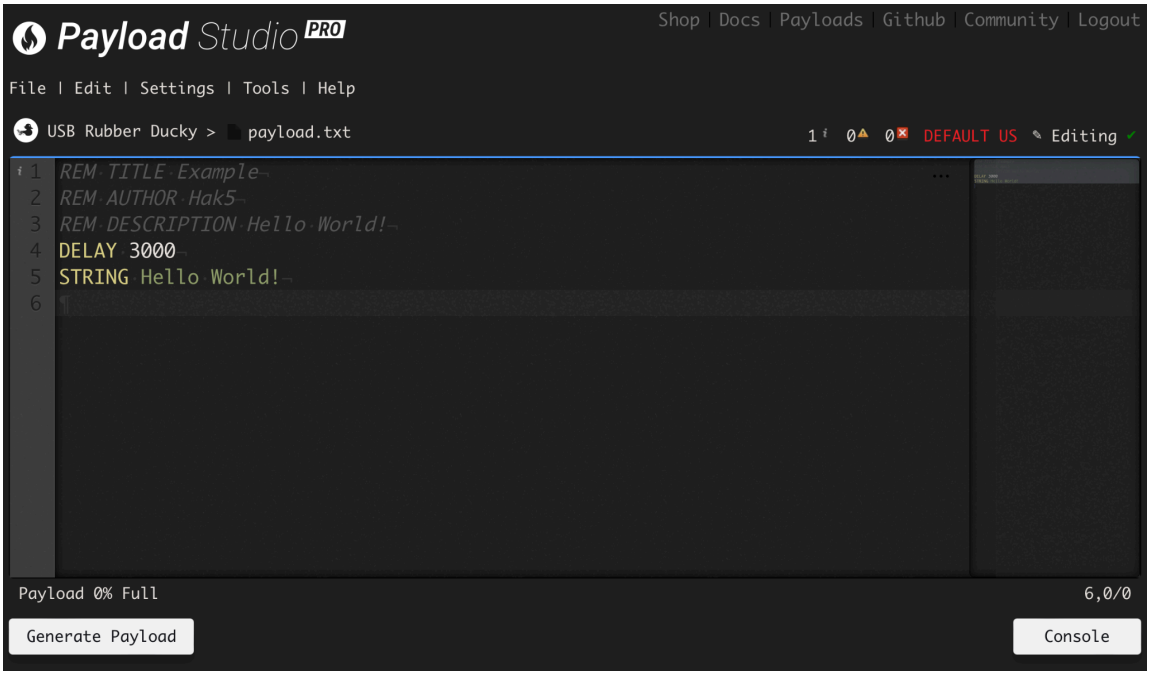
We have formatted this document in the standard IEEE format with clear lab instructions to maintain Reproducibility if this document gets sited in the future.

X. True Results (Experiment)

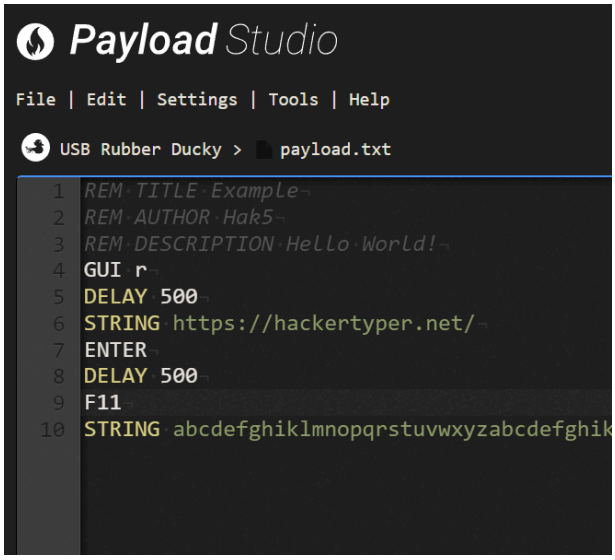
The lab experiment we chose to do was to write a simple BAD USB script using the Hak5's Duckyscript in Payload studio. Our script is harmless and opens up a safe website. But we wanted to showcase how easily this can be done. It is also important to note that you must only conduct A BADUSB attack in controlled, private environments where you have explicit permission. Since this activity without prior authorization is very much illegal. Or could damage someone's computer.

1. Order your BAD USB device
    a. For the purposes of this lab we will be using a Flipper Zero and a Rubber Ducky. In theory you can use any usb device if configured correctly.
    b. https://shop.hak5.org/products/usb-rubber-ducky
    c. https://shop.flipperzero.one

2. Write your own bad USB Script
    a. For this lab we will be using Payload Studio since the Rubber Ducky uses a compiled bin file. But other devices such as the Flipper only need a standard .txt file that can be written in notepad or VS Code.



3. You then write your script

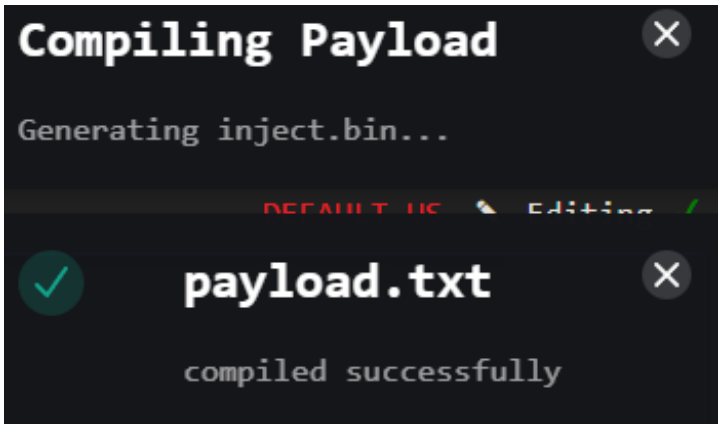


CODE:

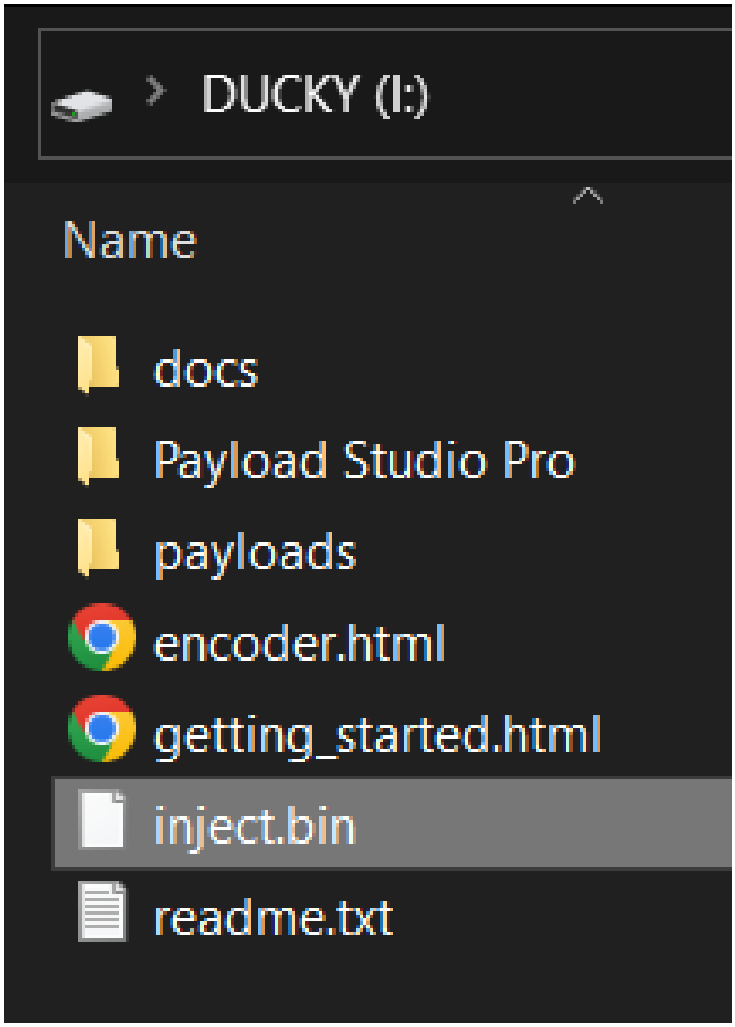
```
GUI r
DELAY 500
STRING https://hackertyper.net/
ENTER
DELAY 500
F11
STRING
abcdefghiklmnopqrstuvwxyzabcdefghiklmnopqrstuvwxyzabcdefghi
klmnopqrstuvwxyz
```

- "GUI r" simulates us the ducky pressing the windows key and r to open the windows "Run" program
- "DELAY 500" makes the script wait 500 milliseconds (0.5 sec) to allow time for the app to load
- "https://hackertyper.net/" the ducky types that URL into run and opens the link in the user's default web browser. This is why we used the run method so there is no issues finding which browser they have installed.
- "ENTER" simulates us the ducky pressing the enter key
- "DELAY 500" This allows time for the website to load. In case the user has slow internet.
- "F11" simulates us the ducky pressing the F11 key to fullscreen the website tab
- "STRING abc…" simulates the ducky pressing random keys on the keyboard for an extended period of time.
- Finally the script ends

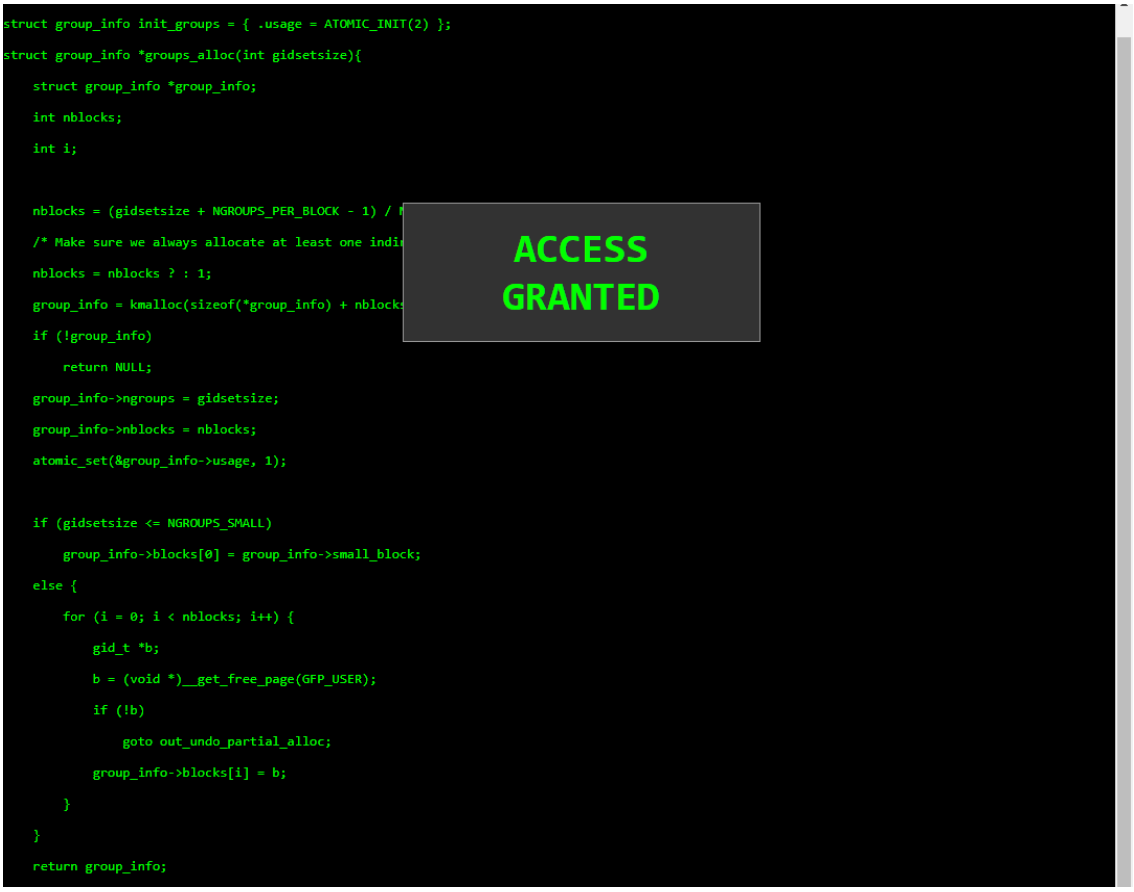
4. We then compile the code into a .bin file in payload studio



5. We then drag and drop our payload.bin file onto the ducky



6. We then disconnect the ducky from our computer. Press the secret arming button (To arm the payload) and now it is ready to plug in to the victim's PC



XI. Analysis

One thing we noticed while researching and experimenting with our BAD USB devices was the fact that they are so easy to use. Once we learned how they functioned and learned the correct syntax of the ducky script. We got the script typed and compiled in under 15 minutes. Granted we did use a quite simple script but the lab was just to showcase how simple it is to write a script and how quickly it can become malicious.

- **IEEE Research Question 1 Answered: "**How damaging can a Bad USB attack be?"
  - Based on our research a Bad USB attack can be very damaging. It can install a backdoor, wipe all the data from a HDD, steal customer data and more.

- **IEEE Research Question 2 Answered:** "How effective are BAD USB Attacks"
  - Based on our research a Bad USB attack. The dubber ducky can clone your Keyboard hardware ID and tricks the host PC into thinking you are typing. This means the Ducky has the same level of access as you do on your computer. This can lead to very damaging outcomes as mentioned above.

XII. Conclusions

In conclusion, the cybersecurity project involving a rubber ducky and a flipper zero was a valuable learning experience for understanding the capabilities and risks associated with BadUSBs. Through the lab experiments, we gained hands-on experience with different types of BadUSB attacks, including keystroke injection and device impersonation, and explored various techniques for detecting and mitigating these attacks.

We also learned that BadUSBs can be challenging to detect and prevent due to their ability to mimic legitimate USB devices and evade traditional security measures. However, hardware-based solutions such as the flipper zero and software-based solutions such as endpoint protection software can be effective in mitigating BadUSB risks.

Overall, this project highlighted the importance of USB device security and the need for ongoing research and development to address the evolving threat landscape. By increasing awareness and understanding of BadUSBs, we can take proactive steps to protect our systems and critical infrastructure from potential attacks.

XIII. Future works

a. AI generating BadUSB scripts on the fly:

   AI is increasingly growing within our world and the tech industry. With the increasing capabilities that AI has been generated, it is definitely possible to say that AI machines will continue to get better at generating scripts for BadUSB users that may want to use them in a malicious way. It is exciting but also scary to see how much AIs will grow in the next few years. Also, what it will mean for BadUSBs and malicious cyber criminals that will use AI to advantage.

b. Future penetration testing devices:

   This project went over an overview of what USBs are and what BadUSBs are and how they work. In this project, we created a lab that involved a Rubber Ducky and a Flipper One device. However, we would like to explore further penetration testing devices that would future advance our penetration testing knowledge. We would like to research more on what different devices are good for BadUSBs and how they work. We do believe that the two devices we did showcase in our lab experiment definitely showed some great ways that a BadUSB script can be implemented on a computer system.

c. Interview someone in the pen-testing filed:

   Interviewing someone who works in the penetration testing field can be an excellent way to gain insights and knowledge about BadUSB and related topics. We would gain an insight on first hand experience from a professional, learn

about the industry's best practices, it would be good for networking opportunities, and build credibility in a research project.

Overall, intervening with someone who works in the penetration testing field can provide invaluable insights, and knowledge for researching BadUSB and related topics. It will help expand our network and establish credibility in our research.

d. Awareness ad campaign:

Spreading awareness and conducting ad campaigns can be an effective way to raise awareness about the dangers of BadUSB and promote best practices for preventing and protecting against such attacks. Increasing awareness can help raise awareness among the general public and teach individuals and organizations to take action to protect against BadUSB attacks. It would also encourage research into BadUSB and related topics by highlighting risks associated with not taking them seriously. Finally, it can also help advocate for policies and regulations that promote better cybersecurity practices and protect against BadUSB attacks. Overall, spreading awareness and conducting ad campaigns can be an effective way to promote research into BadUSB and related topics.

XIV. **References (Do not add just web link use papers as much as possible other can be web based reliable references)**

1. Hak5, “USB rubber ducky,” *Hak5*. [Online]. Available: https://shop.hak5.org/products/usb-rubber-ducky. [Accessed: 23-Apr-2023].
2. “Hak5 Payloads Studio,” *Hak5 PayloadStudio*. [Online]. Available: https://payloadstudio.hak5.org/. [Accessed: 23-Apr-2023].
3. “DuckyScript™ Quick Reference,” *DuckyScript™ Quick Reference - USB Rubber Ducky*. [Online]. Available: https://docs.hak5.org/hak5-usb-rubber-ducky/duckyscript-tm-quick-reference. [Accessed: 23-Apr-2023].
4. “This makes hacking too easy - flipper zero,” *YouTube*, 09-Apr-2023. [Online]. Available: https://youtu.be/nLIp4wd0oXs. [Accessed: 23-Apr-2023].
5. “Do not plug this USB in! – hak5 Rubber Ducky,” *YouTube*, 14-Nov-2022. [Online]. Available: https://youtu.be/kfaHJwcG2mg. [Accessed: 23-Apr-2023].
6. communications@manageengine.com ManageEngine, “BADUSB attack prevention,” *What is BadUSB | How to Protect Against BadUSB Attacks - ManageEngine Device Control Plus*. [Online]. Available: https://www.manageengine.com/device-control/badusb.html. [Accessed: 23-Apr-2023].
7. “Defending against malicious USB firmware with goodusb.” [Online]. Available: https://www.cise.ufl.edu/~butler/pubs/acsac15.pdf. [Accessed: 24-Apr-2023].
8. “Bad USB,” *Flipper Zero - Documentation*. [Online]. Available: https://docs.flipperzero.one/bad-usb. [Accessed: 23-Apr-2023].
9. Stefan, “How bad usbs work,” *Spacehuhn Blog*, 31-Aug-2022. [Online]. Available: https://blog.spacehuhn.com/how-bad-usbs-work. [Accessed: 23-Apr-2023].
10. “BADUSB keystroke injection cable,” *zSecurity*, 06-Apr-2023. [Online]. Available: https://zsecurity.org/product/badusb-keystroke-injection-cable/. [Accessed: 23-Apr-2023].
11. G. Karantzas, “Forensic log based detection for keystroke injection ‘Badusb’ attacks,” *arXiv.org*, 09-Feb-2023. [Online]. Available: https://arxiv.org/abs/2302.04541. [Accessed: 23-Apr-2023].
12. “5-4 studies on countermeasures for malicious USB devices - NICT.” [Online]. Available: https://www.nict.go.jp/publication/shuppan/kihou-journal/journal-vol63no2/journal-vol63no2-05-04.pdf. [Accessed: 24-Apr-2023].
13. Kkamagui, “KKAMAGUI/iron-HID: Iron-HID: Create your own bad USB device (presented at hitbsecconf 2016),” *GitHub*. [Online]. Available: https://github.com/kkamagui/IRON-HID. [Accessed: 23-Apr-2023].
14. “Don't plug it in! how to prevent a USB attack,” *PCMAG*. [Online]. Available: https://www.pcmag.com/how-to/dont-plug-it-in-how-to-prevent-a-usb-attack. [Accessed: 23-Apr-2023].
15. “Securely transferring data from BadUSB devices.” [Online]. Available: https://rocys.ici.ro/documents/43/2020_fall_article_2.pdf. [Accessed: 24-Apr-2023].