Flipper Zero Presentation: Bad Usb

**What is a bad usb:**

* A bad usb is a programmable usb that is used to execute a script when plugged into a device.
* The bad usb is often used to run malicious programs when plugged into a device such as a computers usb ports or even a phone.
* The reason bad usb’s are common threat is that devices connected to a computer are optimized to recognize and connect with various physical devices that are plugged in.
* This is to make things convenient for the end user which comes at the cost of security. As many antivirus’s do not reach the firmware level, these bad usb’s are an easy method of injecting malware unlike executables that are often detected by antivirus’s or filtered by firewalls.
* Bad usb’s are also difficult to successfully use for malicious intents as the person must physically input the bad usb into a port without being detected and likely retrieve it if they don’t wish to be caught.
* Bad usb’s also have a variety of use cases that aren’t as nefarious. As the device can run scripts portably and can type thousands of words per minute, using one of these bad usb’s can make things more efficient.
* A pen tester could use it in their line of work or an IT professional setting up computers.
* Personal use cases are many, but the scope of this presentation is the use of bad usb’s in conjunction with the flipper zero.

**How does it work?**

Bad usb’s work by changing the microcontroller in a normal usb to make the usb register as a human interface device such as a keyboard, camera, modem, or other firmware that will give the device access where it is not intended. Often the usb is used as a keyboard with a prewritten text script which will execute on connection. For example, a bad usb could log keystrokes to open a command line interface at which point a script could be run. The compromised device is now at risk of everything from data theft, and ransomware but can go much further compromising the network and adjacent devices.

**Protecting against BadUSB:**

Protecting against bad usb attacks aren’t very easy. As an individual making sure device ports are always secure and usb’s that are inputted aren’t dangerous is necessary. For larger networks such with many devices laying around and shared workstations, usb injected malware is harder to prevent. Blocking all usb ports isn’t a feasible solution therefore limiting access within a within a computer and intra-network security is a priority. Checking hardware ID’s and only using trusted devices is common practice. Other precautions commonly taken to protect against these kinds of attacks are always having a login to access a computer, port inspection before logging in and even physically restricting who can enter into a building/ be around electronic devices.

A picture containing text

Description automatically generated

<https://www.bleepingcomputer.com/news/security/fbi-hackers-use-badusb-to-target-defense-firms-with-ransomware/> (Link to image above)

**How microcontrollers are changed in a usb to turn it rogue:**

Not all usb’s are equipped with the necessary hardware to be transformed into a bad usb.

To reconfigure the usb into a human interface device - a.k.a. a bad usb capable of interacting with computers once plugged in – the steps are as followed :

/\* more detailed explanations necessary \*/

* Getting a usb with a viable microcontroller
* Programming it to register as a keyboard: dumping firmware and flashing keyboard image
* Compiling custom firmware (uses C)
* Writing a script using duckyscript
* Injecting payload via usb port

<https://github.com/brandonlw/Psychson>

<https://null-byte.wonderhowto.com/how-to/make-your-own-bad-usb-0165419/>

**Cases where a badusb was used:**

* Stuxnet
* Fin7 Gift usb
* DarkVishnya bash bunny attacks

**Flipper zero context:**

The bad usb feature is one of the many features of the Flipper zero. The bad usb uses duckyscript, a language common for scripting commands to a device and the qflipper application used to update the flipper firmware. Duckyscript is written in a text file and begins its execution once uploaded to the flipper and the badusb feature is selected. The bad usb feature is widely used and many scripts can be found online on github and other open source code platforms. This makes the flipper zero dangerous as malicious hacking is accessible to non-programmers and anyone intent on committing cybercrime. The flipper zero can act as a bad usb but has increased functionality. Multiple scripts can be saved on the flipper and it’s as simple as connecting the flipper via a usb to a port, selecting a payload, and clicking run. These hacking tools are often controversial and a topic of debate as they’re marketed as consumer products , however the field of cybersecurity is always expanding making “hacking” more difficult then running a duckyscript. The flipper zero and similar devices are always in circulation, often being used by pen testers to make devices and technology in general more secure.

**Reverse Shell:**

**Duckyscript:**

\*Enter Ducky script\*

**Questions:**

/\*Can a bad usb access root/ execute sudo commands? \*/

\*/ Different microcontrollers/ what is necessary \*/