Shuck the Store-Bought: Grow Your Own "HaKC5" Gear

Problem statement and solution proposal

(hakc5) ~ whoami

Name: Bill Swearingen
Date of Birth: 22 November, 1987
SSN: (380)-444-3233

Co-Founder, SecKC
Co-Founder, CCCKC
Co-Founder, Trifident Security Advisory

First time Corncon attendee, I am here on my own free will. These views are not my own. They are my company's, my sponsors', and possibly Hak5's lawyers.

HAK5

TRUST YOUR TECHNOLUST

PACKET SQUIRREL

A matchbook-sized linux box packing multiple network payloads - like packet sniffing, VPN tunneling and man-in-the-middle attacks.

BASH BUNNY

A quad-core Linux-box-on-USB-stick mimicking multiple trusted devices to deploy advanced pentest and IT automation payloads.

USB RUBBER DUCKY

A "flash drive" that types keystroke injection payloads into unsuspecting computers at incredible speeds. As seen on Mr. Robot.

LAN TURTLE

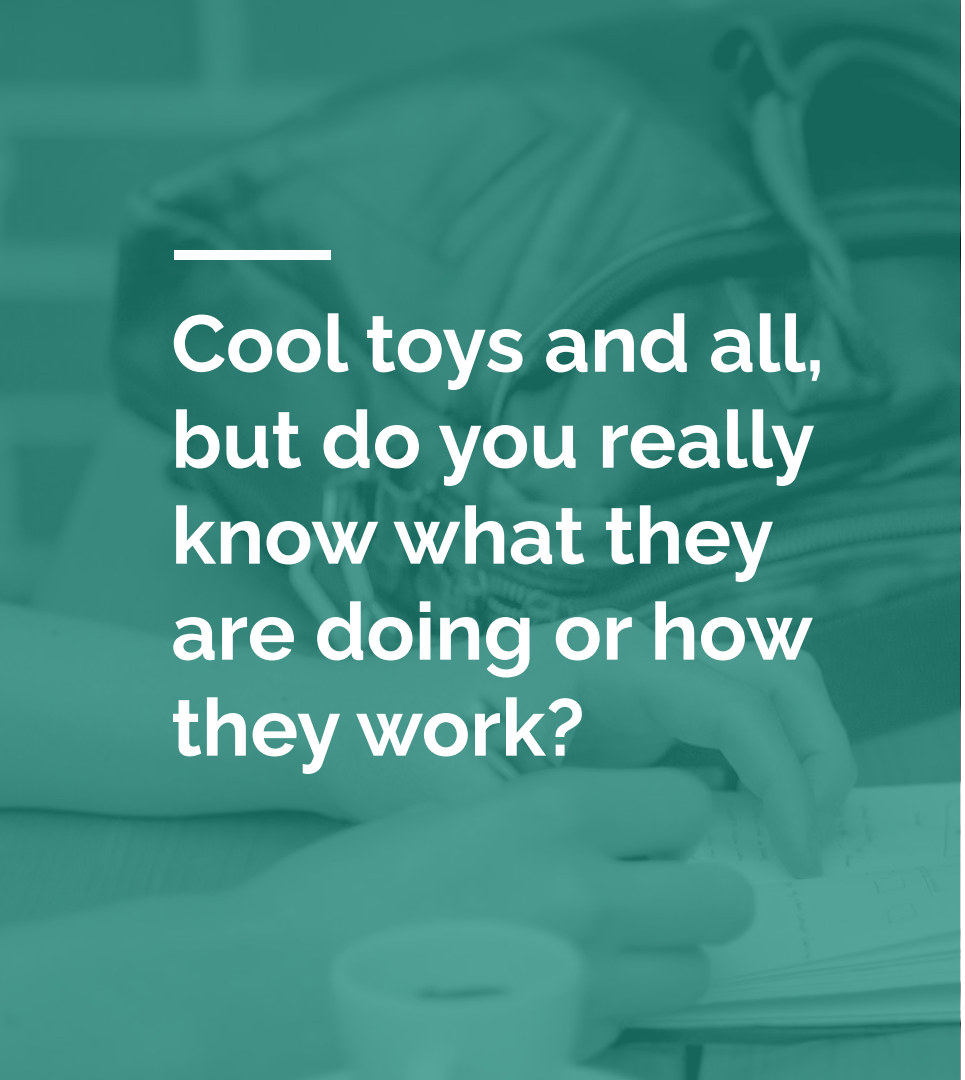
A Remote Access Toolkit posing as an ordinary USB Ethernet adapter. Drop it on a LAN for an instant backdoor shell.

SHARK JACK

Jack into a network and instantly run advanced recon, exfiltration, attack and automation payloads. Available battery or USB-C powered.

PAGER

The next-gen standalone pentest experience. With a vibrant display, on-device controls, battery and modern radios.



Cool toys and all, but do you really know what they are doing or how they work?

You should build your own.

Hands-on Learning

Cheaper, flexible, and open source

Improve on the design, give back to the community

A quick primer on USB

The USB human interface device class (USB HID class) is a part of the USB specification for computer peripherals: it specifies a device class (a type of computer hardware) for human interface devices such as keyboards, mice, touchscreen, touchpad, game controllers and alphanumeric display devices.

Enumeration

The computer sees something new on the USB bus and asks: “Who are you?”

USB Descriptors

The device replies with small data structures called USB descriptors. These describe:
Vendor ID (VID): who made it
Product ID (PID): which model it is
Device class: what kind of thing it is (keyboard, mouse, storage, audio, etc.)
Endpoints: little “channels” the device uses to send or receive data

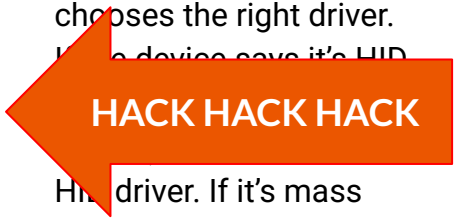
Driver

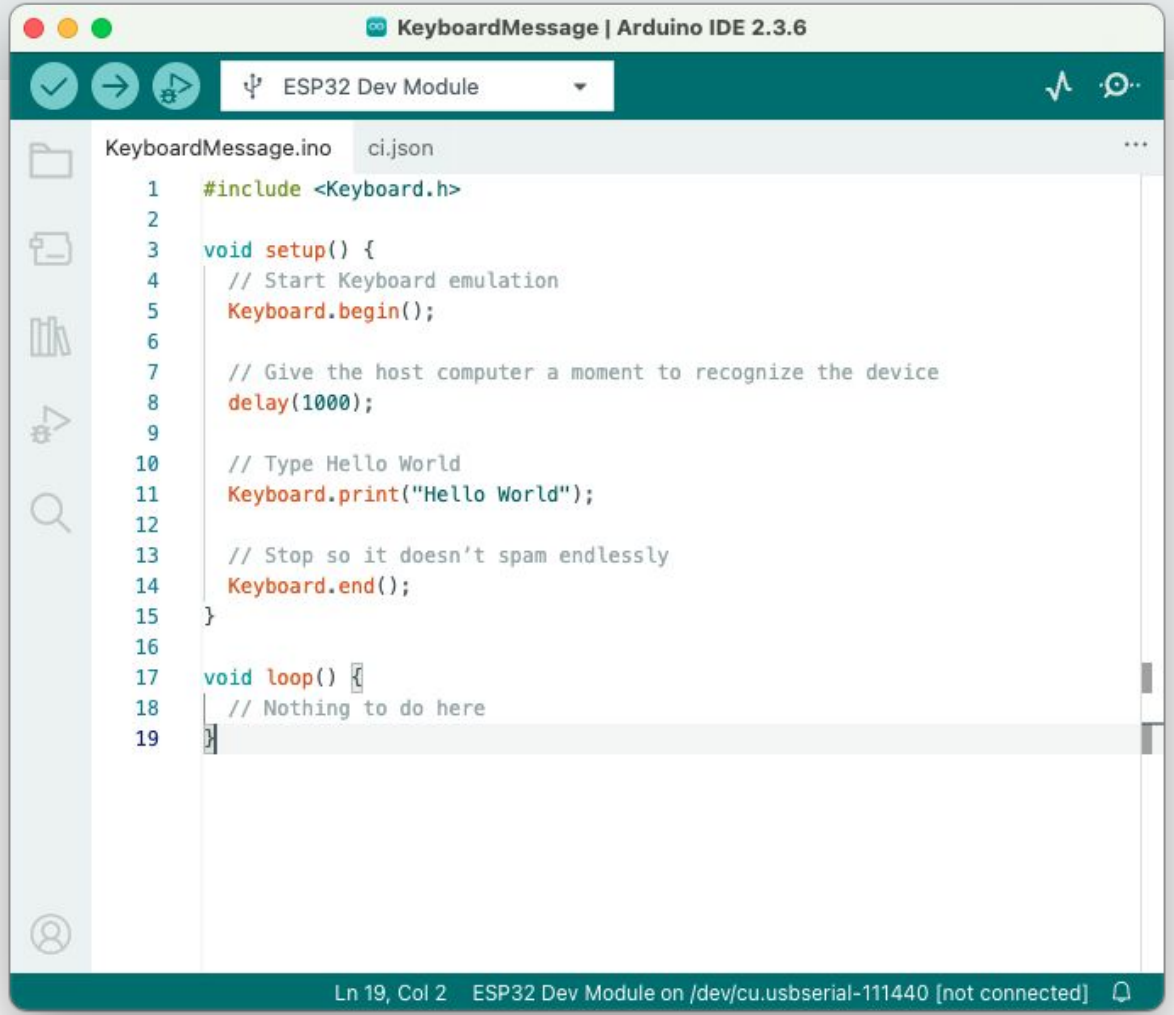
The operating system looks at that info and chooses the right driver. If the device says it’s HID (Human Interface Device), the OS loads the HID driver. If it’s mass storage, it loads the storage driver.

Configuration

Once the driver is picked, the OS sets configuration values and the device is ready to go. From that point on, the OS knows whether it should expect keystrokes, audio streams, files, or whatever the device claimed it could do.

Enumeration	USB Descriptors	Driver	Configuration
The computer sees something new on the USB bus and asks: “Who are you?”	The device replies with small data structures called USB descriptors. These describe: **Vendor ID (VID):** who made it **Product ID (PID):** which model it is **Device class:** what kind of thing it is (keyboard, mouse, storage, audio, etc.) **Endpoints:** little “channels” the device uses to send or receive data	The operating system looks at that info and chooses the right driver. [illegible] device says it’s HID [illegible] driver. If it’s mass storage, it loads the storage driver.	Once the driver is picked, the OS sets configuration values and the device is ready to go. From that point on, the OS knows whether it should expect keystrokes, audio streams, files, or whatever the device claimed it could do.




KeyboardMessage | Arduino IDE 2.3.6
ESP32 Dev Module
KeyboardMessage.ino
ci.json
1 #include <Keyboard.h>
2
3 void setup() {
4 // Start Keyboard emulation
5 Keyboard.begin();
6
7 // Give the host computer a moment to recognize the device
8 delay(1000);
9
10 // Type Hello World
11 Keyboard.print("Hello World");
12
13 // Stop so it doesn't spam endlessly
14 Keyboard.end();
15 }
16
17 void loop() {
18 // Nothing to do here
19 }
Ln 19, Col 2 ESP32 Dev Module on /dev/cu.usbserial-111440 [not connected]

Ok super star hackers, what is the problem with that approach?

```
(hakc5) ~ nano ~/Library/Arduino15/packages/esp32/hardware/esp32/3.3.1/boards.txt
```

```
  UW PICO 5.09                    File: /Users/haxxx/Library/Arduino1

##################################################################
### DO NOT PUT BOARDS ABOVE THE OFFICIAL ESPRESSIF BOARDS! ###
##################################################################

# Generic definition to be used for USB discovery of CDC/JTAG
esp32_family.name=ESP32 Family Device
esp32_family.hide=true
#esp32_family.vid.0=0x303a
#esp32_family.pid.0=0x1001
#esp32_family.vid.0=0x046D #(Logitech)
#esp32_family.pid.0=0xC34B #(Logitech USB Keyboard)
esp32_family.upload_port.0.vid=0x303a
esp32_family.upload_port.0.pid=0x1001
esp32_family.build.board=ESP32_FAMILY
```

0x00 – Defined at the interface level (composite devices use this)	**0x0B** – Smart Card
0x01 – Audio (headsets, speakers, mics)	**0x0D** – Content Security (DRM dongles, encryption tokens)
0x02 – Communications (modems, network adapters)	**0x0E** – Video (webcams, capture devices)
0x03 – HID (Human Interface Device: keyboards, mice, game controllers)	**0x0F** – Personal Healthcare (blood pressure monitors, glucose meters)
0x05 – Physical (rare, sensors)	**0x10** – Audio/Video (combined AV devices)
0x06 – Image (scanners, still cameras)	**0x11** – Billboard (USB-C alternate mode negotiation helpers)
0x07 – Printer	**0xDC** – Diagnostic Device
0x08 – Mass Storage (flash drives, external hard drives, memory card readers)	**0xE0** – Wireless Controller (Bluetooth dongles, wireless adapters)
0x09 – Hub (USB hubs themselves)	**0xEF** – Miscellaneous (composite gadgets that don’t fit elsewhere)
0x0A – CDC Data (data side of communications devices)	**0xFE** – Application Specific (DFU bootloaders, IRDA bridges)

What if mass storage is blocked by the device?

Enumeration step:

When the host asks the device who it is, the device can reply with Class = 0x00 and then list separate interface descriptors. Each interface includes its own class code. So one device can say: interface 0 = HID keyboard, interface 1 = Mass Storage, interface 2 = CDC Ethernet, etc.

Why this matters:

If your policy or USB controller decides only from the device class field and treats 0x08 as blocked and 0x03 as allowed, a composite device that advertises 0x00 can slip through because the decision was never made on the per-interface classes. The OS will still load the appropriate drivers for each interface unless the host enforces per-interface blocking.


HACK
HACK
HACK

Ok, why you should make your hacking equipment yourself.

HAK5

TRUST YOUR TECHNOLUST

USB RUBBER DUCKY

$100.00

NEW VERSION OF THE BEST SELLING HOTPLUG

With a few seconds of physical access, all bets are off...

USB RUBBER DUCKY

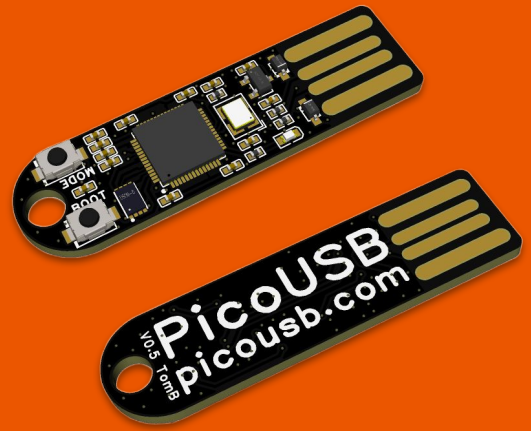
PRO BUNDLE

ELITE BUNDLE

Qty

— 1 +

ADD TO CART



PicoUSB -Raspberry Pi Pico RP2040 powered Bad USB (Rubber Ducky)

★★★★★ 7 Reviews | Add Your Review

~~$12.99~~ **$7.99**

AVAILABILITY: **IN STOCK**

SKU: **CQA240517P**

WEIGHT: **60G**

Buy 2 for $4.99 each and **save 38%**

Buy 5 for $3.99 each and **save 50%**

Buy 10 for $2.99 each and **save 63%**

PicoUSB: Raspberry Pi (Pico) RP2040 powered Bad USB (Rubber Ducky)

When inserted in a PC, it acts as a keyboard, a mouse or volume control knobs, or all of these together, all in one device.

Here are some advantages of it:

- Dual sided USB - you can plug it both ways, you don't have to worry if you oriented it correctly.
- Easy Programming - you are able to use easy pseudo-code to program it.
- Edit Mode Button for easy setup:
  - If pressed - does not execute code and opens PicoUSB as mass storage for easy editing.
  - Not pressed - code executes and does not show as a mass storage device for more covert operation.

Open Source - Even if you don't want to buy the PicoUSB, you can still use the firmware for your Raspberry Pi Pico.

Versatile - Raspberry Pi RP2040 based so you can write your own code.

HAK5

TRUST YOUR TECHNOLUST

The O.MG Cable is a hand made USB cable with an advanced implant hidden inside. It is designed to allow your Red Team to emulate attack scenarios of sophisticated adversaries.

Until now, a cable like this would cost $20,000 (ex: NSA's COTTONMOUTH-I). These cables will allow you to test new detection opportunities for your defense teams. They are also extremely impactful tools for teaching and training.

$180.00

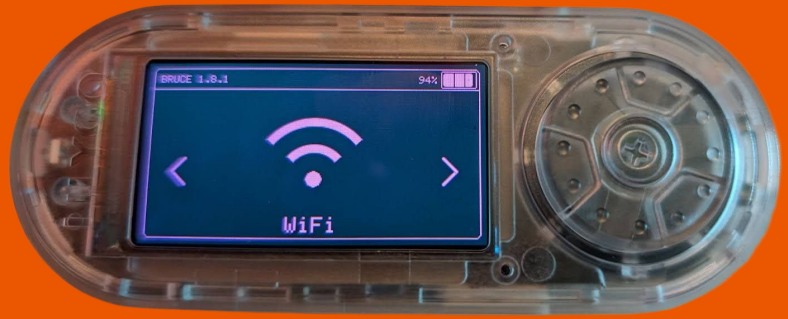
$41.00

Evil Crow Wind – USB-C WiFi Enabled BadUSB Device

Evil Crow Cable Wind is a BadUSB device based on ESP32-S3 (It only allows charging of the mobile phone).

Evil Crow Cable Wind can be controlled with a web panel over Wi-Fi, the device is configured in STATION mode. You will need to set up a Wi-Fi access point with your mobile phone or another device, Evil Crow Cable Wind will automatically connect to it.

Evil Crow Cable Wind is pre-configured with English layout (EN_US), but is compatible with other keyboard layouts



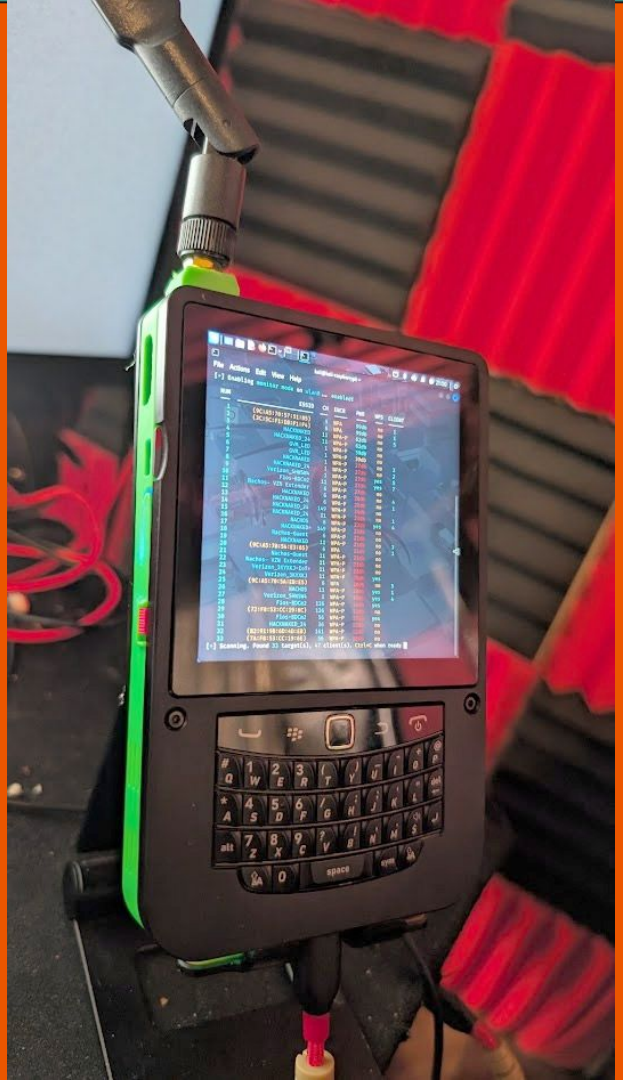
$55.00

LilyGo T-Embed CC1101 device running the Bruce firmware

The CC1101 allows for Sub-GHz communications, and the hardware supports Wifi (2.4GHz), BLE, IR, and more.

Bruce is meant to be a versatile ESP32 firmware that supports a ton of offensive features focusing on facilitating Red Team operations. It also supports m5stack products and works great with Cardputer, Sticks, M5Cores, T-Decks and T-Embeds.

Wifi Attacks (Beacon Spam, Deauth, Evil Portal, Sniffer) **Bluetooth Attacks** (Bad BLE, BLE Keyboard, Ble Spam, Scanner) **RFID Attacks** (Clone Tag, Kill Tag, Emulate) **IR Attacks** (TV-B-Gone, IR Receiver, Emulator) **FM Attacks** (Broadcast, Spectrum Analyzer)



Hackberry Pi CM5 9900 cyberdeck

This Hackberry Pi CM5 9900 cyberdeck, customized with an NVME SSD running Kali, enhanced by an external antenna, is a compact, responsive hacking platform. It's ideal for pentesters, infosec enthusiasts, and anyone craving flexible Linux power in a handheld package. The modularity means you can keep upgrading: swap CM5 modules, change SSDs, or experiment with sensors via the Stemma I2C port.

https://eclypsium.com/blog/build-the-ultimate-cyberdeck-hackberry-pi/

$170.00

Bad Camera

Eclypsium researchers gave a talk at Defcon 33 showing how to convert a Lenovo 510 FHD Webcam into a USB Gadget and implemented BadUSB-style attacks (i.e., Rubber Ducky, O.MG Cable, Bash Bunny).

https://eclypsium.com/blog/badcam-now-weaponizing-linux-webcams/

$50.00

LILYGO T-Dongle S3 w/USBArmyKnife

USB HID attacks, mass storage emulation, network device impersonation and WiFi/Bluetooth exploits.

Complete control over how and when your payloads are run. Plug in and execute, leave behind and trigger over WiFi, run on a timer or build a Hollywood-esq UI. Manage and deploy your attacks effortlessly using just a phone using a user-friendly Bootstrap web interface.

Want more? Deploy the agent and execute commands even when the machine is locked. Working over the serial interface egress is incredibly hard to detect. You can even view the victims screen over the devices' dedicated WiFi connection.

$17.00

ULTRA CHEAP RUBBER DUCKY DEMONSTRATION

Everyone, please join me in a prayer to the demonstration gods….

(hakc5) ~ echo "Thank You CornCon"

SUPER THANKS TO THE CORNCON CREW, ESPECIALLY CHRIS COOPER.

Please check out SecKC

Please check out SecDSM

Please try all this at home.