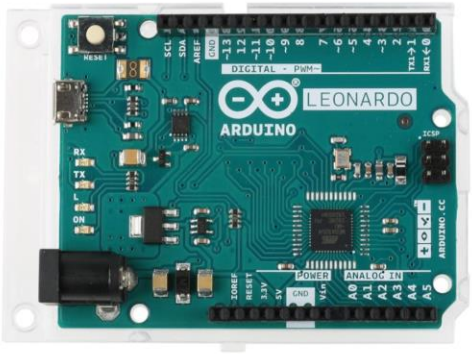
Rubber Ducky E-Mailer

https://github.com/zkbyqd/rubber-ducky-emailer

Inhalt

- Welche Hardware, warum und welche ist tauglich?
- Allgemeine Definition
- Was passiert genau?
- Beispiel einer Ausführung
- Payload Möglichkeiten / Was geht noch?
- Real-World Szenarios
- Wie schützt man sich?
- Spezielles meines Payloads
- Bonus

Hardware



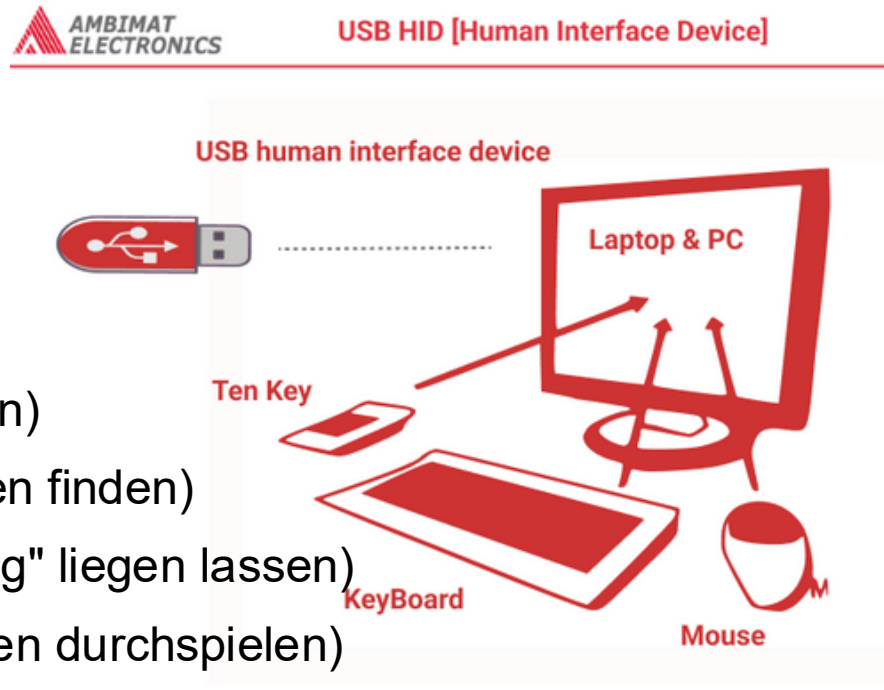
- Arduino Leonardo
  - Hatte ich gerade da
  - Native USB-Unterstützung
  - Mehr Speicher und bessere Performance als ein Digispark

Welche Geräte sind generell tauglich?

- USB-HID fähige Microcontroller
- Rubber Ducky von Hak5
- Umgebauter, nicht USB-HID fähiger, Microcontroller

Was ist ein Rubber Ducky?

- Ein USB-Gerät, das vorgibt ein "Human Interface Device" zu sein z.B. eine Tastatur
- Es führt automatisch Tastatureingaben aus, sobald es eingesteckt wird
- Das Betriebssystem vertraut Eingabegeräten, daher findet keine Sicherheitsabfrage statt

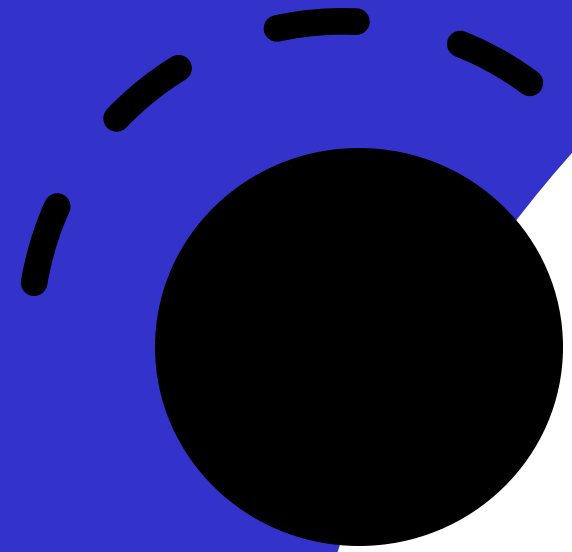


Einsatzgebiete sind:

- Pentests (Sicherheitslücken simulieren)
- Red Teaming (Zugang zu Zielsystemen finden)
- Social Engineering (USB-Stick "zufällig" liegen lassen)
- Malware-Tests (Echte Angriffsszenarien durchspielen)

Was passiert?

- Initialisieren als Tastatur (automatisch)
- Super Key + R drücken
- CMD versteckt öffnen
- Schadskript aus dem Internet laden (bspw. Pastebin/ Github Repo)
- Informationen sammeln
- Informationen über SMTP versenden
- Payload selbstzerstören



Beispiel

Payload Möglichkeiten

- Recon (Systeminfos sammeln)
- Payload Delivery (Schadcode von Server laden und ausführen
- Reverse Shell (Zugang zurück zu unserem System)
- Credential Harvesting (Passwörter auslesen aus Browser, WiFi)
- Persistence (Autostart, geplanter Task, Registry-Key)
- Screencapture / Keylogging (Screenshot vom Desktop oder Keys loggen)
- UserAccountControl-Bypass / PrivEsc (Adminrechte erlangen)
- AntiVirus/Defender-Bypass (Temporäres deaktiveren oder umgehen von AV)

Real-World Szenarios

- Szenario 1
  - USB-Stick "zufällig" liegen lassen
  - Unschuldiger ist neugierig und steck ihn in sein Gerät
  - Daten werden ausgelesen und via E-Mail versandt
  - Payload löscht sicht selbst, sodass man meine E-Mail nicht reversen kann
- Szenario 2
  - Ich stecke meinen USB-Stick bei einem Zielsystem ein
  - Durch Physical-Pentesting in bspw. Personalabteilung
  - Der Payload wird in den Autostart gepackt
  - Lasse Keys loggen, um an eventuelle Credentials zu gelangen

Wie schützt man sich?

- USB-Ports sperren (Group Policy oder BIOS)
- USBGuard (Device Control Software)
- Keine unbekannten USB-Geräte anstecken
- Physische Portblocker

Spezielles meines Payloads

Selbstzerstörung

```python
import os # Zugriff auf das Betriebssystem
try:
    os.remove(__file__) # Löscht das aktuell laufende Skript
except:
    pass
```

Exceptions gekürzt

- Gefährlich
- Fängt alle BaseExceptions
- KeyboardInterrupt (Ctrl + C)

Bonus

- https://hackaday.io/project/202218-hidden-hid-v2-worlds-smallest-rubber-ducky

- "Hidden HID v2 - worlds smallest Rubber Ducky"