Bluetooth Headphone Jacking: A Key to Your Phone

Hacking Airoha-based Bluetooth Audio Devices

ERNW
providing security.

About us

Dennis Heinze
Security Researcher &
Penetration Tester

Frieder Steinmetz
Security Researcher &
Penetration Tester

Team Mobile and IoT Security @ ERNW

Previously:

AURACAST™

Bluetooth Auracast Research.
Goal: Obtain firmware and analyze implementation.

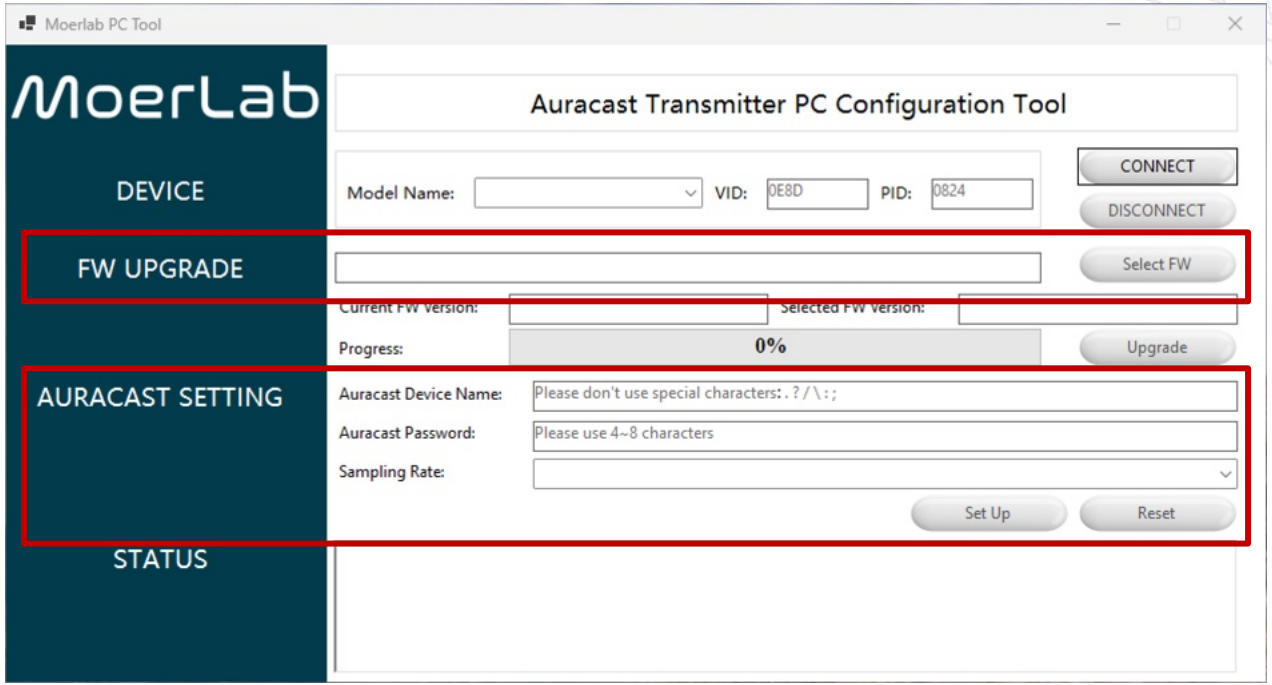
We have a firmware update file for this thing.


```
00000000: bb3b a797 eccc 2084 3272 2c7b c39f bbe8  .;.... .2r,{....
00000010: 6af3 ba00 134d 7a76 3b25 21fe 4012 c002  j....Mzv;%!.@...
00000020: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000030: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000040: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000050: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000060: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000070: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000080: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000090: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000a0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000b0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000c0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000d0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000e0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000f0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000100: 1100 0a00 0201 0010 0000 7016 1300 1300  ..........p.....
00000110: 1c00 5635 2e32 2e32 00ff ffff ffff ffff  ..V5.2.2........
00000120: ffff ffff ffff ffff ffff ffff ffff 1200  ................
00000130: 1c00 0200 0000 0010 0000 00f0 0800 0030  ...............0
00000140: 0100 0000 0900 0040 1500 0020 0e00 1400  .......@... ....
00000150: 4400 0200 0000 494d ad6c 716d f400 ad12  D.....IM.lqm....
00000160: 9a6d 7d21 74d8 ecfd f36d 3889 37a4 b58f  .m}!t....m8.7...
```

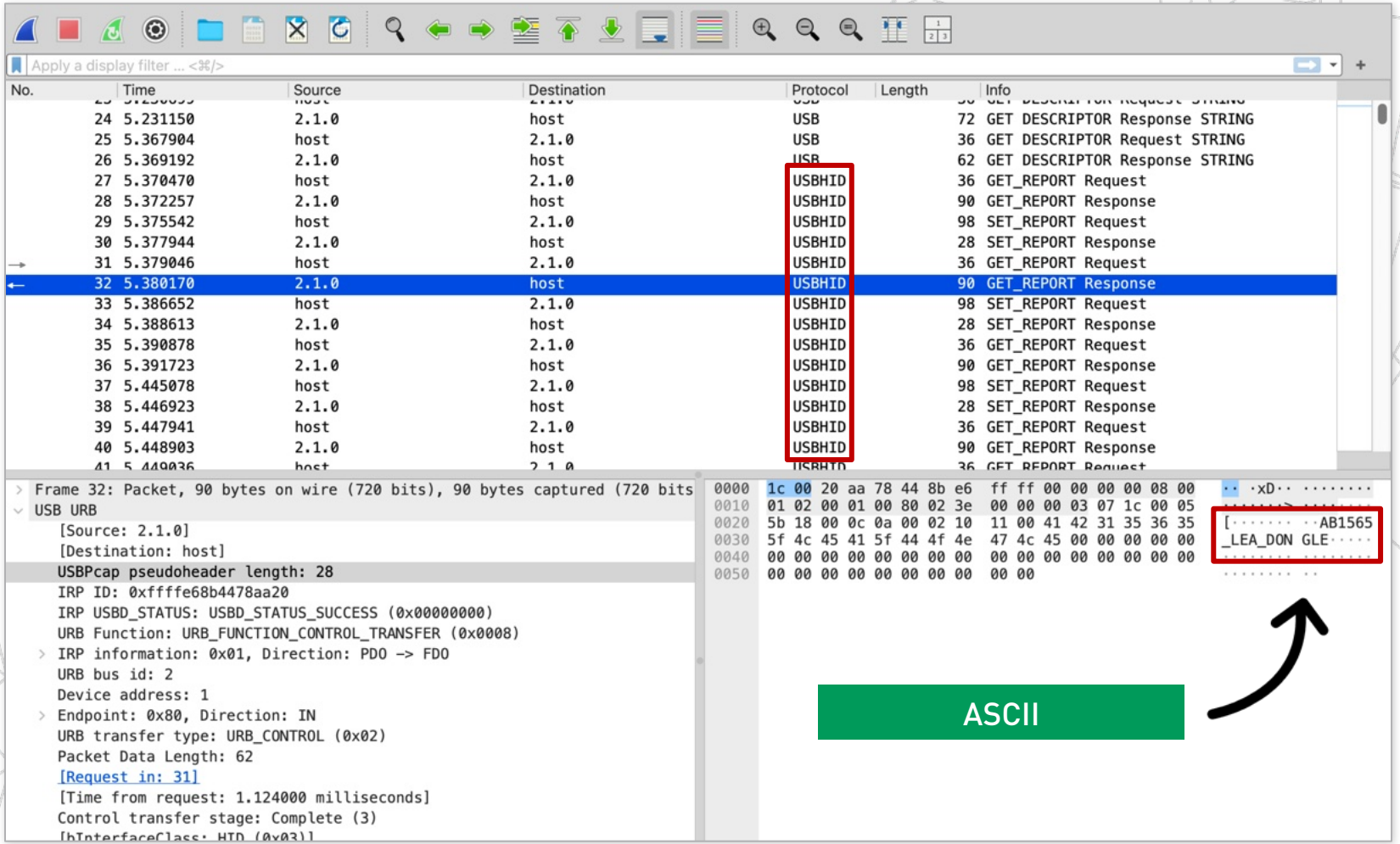
But it's a blob we don't know how to unpack.

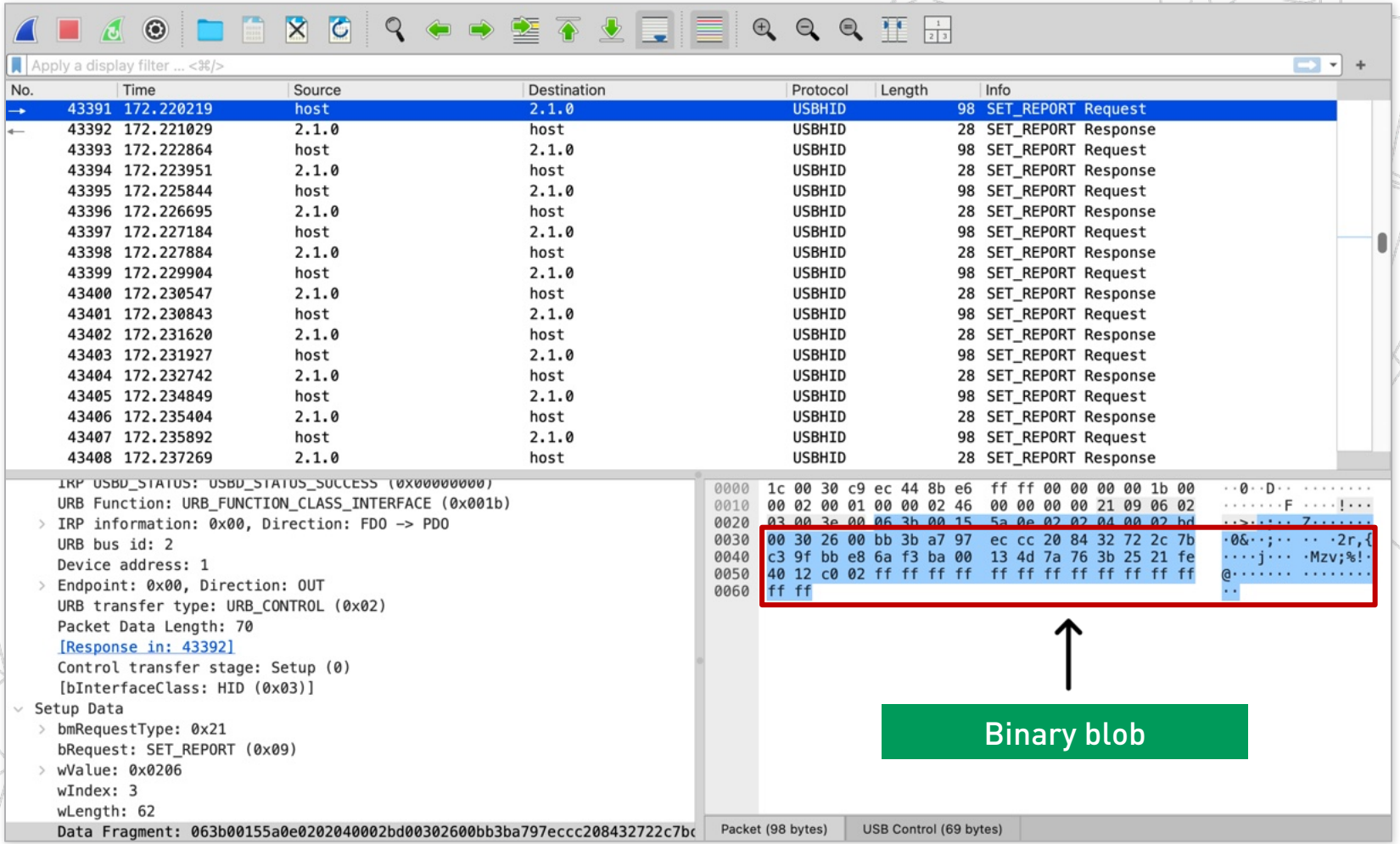
Firmware Update →

Settings →



USB-based tool for configuration and firmware updates.


Apply a display filter ... <⌘/>
No. Time Source Destination Protocol Length Info
24 5.231150 2.1.0 host USB 72 GET DESCRIPTOR Response STRING
25 5.367904 host 2.1.0 USB 36 GET DESCRIPTOR Request STRING
26 5.369192 2.1.0 host USB 62 GET DESCRIPTOR Response STRING
27 5.370470 host 2.1.0 USBHID 36 GET_REPORT Request
28 5.372257 2.1.0 host USBHID 90 GET_REPORT Response
29 5.375542 host 2.1.0 USBHID 98 SET_REPORT Request
30 5.377944 2.1.0 host USBHID 28 SET_REPORT Response
31 5.379046 host 2.1.0 USBHID 36 GET_REPORT Request
32 5.380170 2.1.0 host USBHID 90 GET_REPORT Response
33 5.386652 host 2.1.0 USBHID 98 SET_REPORT Request
34 5.388613 2.1.0 host USBHID 28 SET_REPORT Response
35 5.390878 host 2.1.0 USBHID 36 GET_REPORT Request
36 5.391723 2.1.0 host USBHID 90 GET_REPORT Response
37 5.445078 host 2.1.0 USBHID 98 SET_REPORT Request
38 5.446923 2.1.0 host USBHID 28 SET_REPORT Response
39 5.447941 host 2.1.0 USBHID 36 GET_REPORT Request
40 5.448903 2.1.0 host USBHID 90 GET_REPORT Response
Frame 32: Packet, 90 bytes on wire (720 bits), 90 bytes captured (720 bits
USB URB
[Source: 2.1.0]
[Destination: host]
USBPcap pseudoheader length: 28
IRP ID: 0xffffe68b4478aa20
IRP USBD_STATUS: USBD_STATUS_SUCCESS (0x00000000)
URB Function: URB_FUNCTION_CONTROL_TRANSFER (0x0008)
IRP information: 0x01, Direction: PDO -> FDO
URB bus id: 2
Device address: 1
Endpoint: 0x80, Direction: IN
URB transfer type: URB_CONTROL (0x02)
Packet Data Length: 62
[Request in: 31]
[Time from request: 1.124000 milliseconds]
Control transfer stage: Complete (3)
0000 1c 00 20 aa 78 44 8b e6 ff ff 00 00 00 00 08 00
0010 01 02 00 01 00 80 02 3e 00 00 00 03 07 1c 00 05
0020 5b 18 00 0c 0a 00 02 10 11 00 41 42 31 35 36 35
0030 5f 4c 45 41 5f 44 4f 4e 47 4c 45 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00
ASCII


Apply a display filter ... <⌘/>
No. Time Source Destination Protocol Length Info
43391 172.220219 host 2.1.0 USBHID 98 SET_REPORT Request
43392 172.221029 2.1.0 host USBHID 28 SET_REPORT Response
43393 172.222864 host 2.1.0 USBHID 98 SET_REPORT Request
43394 172.223951 2.1.0 host USBHID 28 SET_REPORT Response
43395 172.225844 host 2.1.0 USBHID 98 SET_REPORT Request
43396 172.226695 2.1.0 host USBHID 28 SET_REPORT Response
43397 172.227184 host 2.1.0 USBHID 98 SET_REPORT Request
43398 172.227884 2.1.0 host USBHID 28 SET_REPORT Response
43399 172.229904 host 2.1.0 USBHID 98 SET_REPORT Request
43400 172.230547 2.1.0 host USBHID 28 SET_REPORT Response
43401 172.230843 host 2.1.0 USBHID 98 SET_REPORT Request
43402 172.231620 2.1.0 host USBHID 28 SET_REPORT Response
43403 172.231927 host 2.1.0 USBHID 98 SET_REPORT Request
43404 172.232742 2.1.0 host USBHID 28 SET_REPORT Response
43405 172.234849 host 2.1.0 USBHID 98 SET_REPORT Request
43406 172.235404 2.1.0 host USBHID 28 SET_REPORT Response
43407 172.235892 host 2.1.0 USBHID 98 SET_REPORT Request
43408 172.237269 2.1.0 host USBHID 28 SET_REPORT Response
URB Function: URB_FUNCTION_CLASS_INTERFACE (0x001b)
IRP information: 0x00, Direction: FDO -> PDO
URB bus id: 2
Device address: 1
Endpoint: 0x00, Direction: OUT
URB transfer type: URB_CONTROL (0x02)
Packet Data Length: 70
[Response in: 43392]
Control transfer stage: Setup (0)
[bInterfaceClass: HID (0x03)]
Setup Data
bmRequestType: 0x21
bRequest: SET_REPORT (0x09)
wValue: 0x0206
wIndex: 3
wLength: 62
Data Fragment: 063b00155a0e0202040002bd00302600bb3ba797eccc208432722c7bc
0000 1c 00 30 c9 ec 44 8b e6 ff ff 00 00 00 00 1b 00
0010 00 02 00 01 00 00 02 46 00 00 00 00 21 09 06 02
0020 03 00 3e 00 06 3b 00 15 5a 0e 02 02 04 00 02 bd
0030 00 30 26 00 bb 3b a7 97 ec cc 20 84 32 72 2c 7b
0040 c3 9f bb e8 6a f3 ba 00 13 4d 7a 76 3b 25 21 fe
0050 40 12 c0 02 ff ff ff ff ff ff ff ff ff ff ff ff
0060 ff ff
Binary blob
Packet (98 bytes)
USB Control (69 bytes)

Firmware Update Command?

USB HID Packet

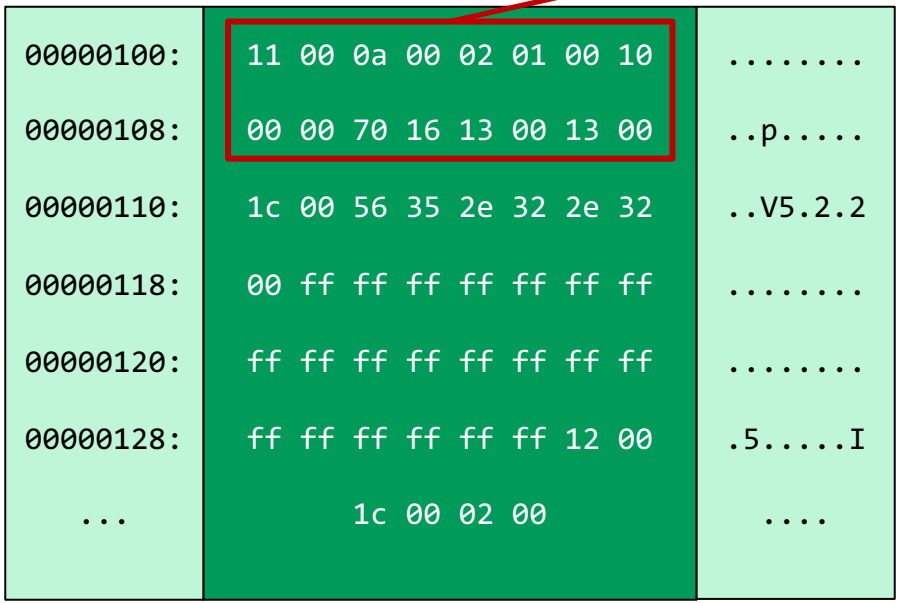
15 5a 0e 02	02 04	00 01 bd	00 30 26 00	bb 3b a7 97 ec cc 20 84 32 72 2c 7b c3 9f ...

00000000:	bb 3b a7 97 ec cc 20 84	.;.... .
00000008:	32 72 2c 7b c3 9f bb e8	2r,{....
00000010:	6a f3 ba 00 13 4d 7a 76	j....Mzv
00000018:	3b 25 21 fe 40 12 c0 02	;%!.@...
00000020:	ff ff ff ff ff ff ff ff	....
00000028:	ff ff ff ff ff ff ff ff	....
...	ff ff ff ff	...

Firmware Update File

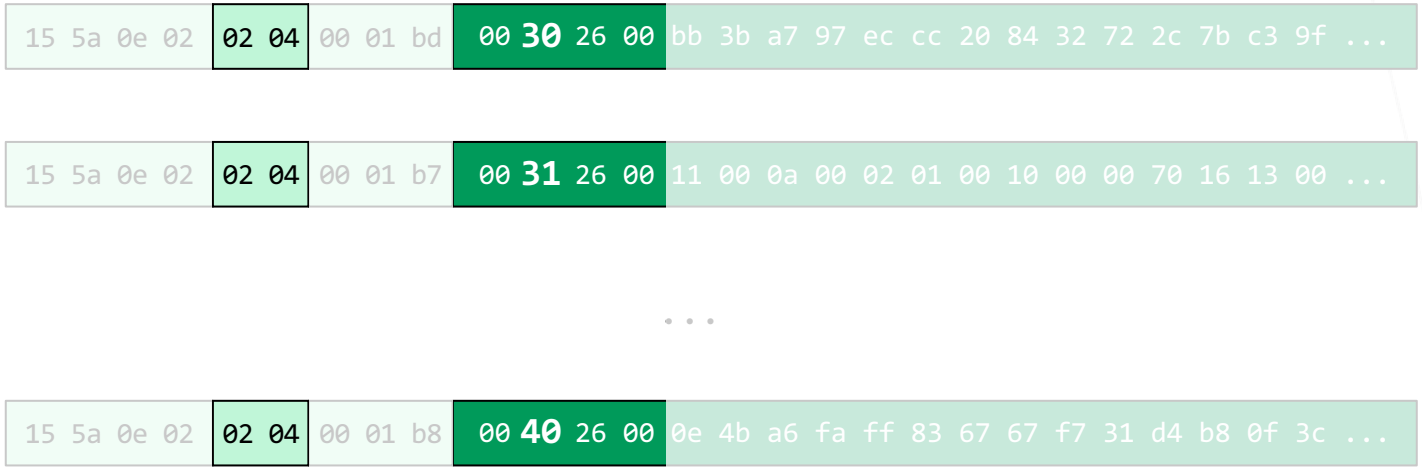
Firmware Update Command?

USB HID Packet



Firmware Update File

Writing Data



In the meantime,...

```
00000000: bb3b a797 eccc 2084 3272 2c7b c39f bbe8  .;.... .2r,{....
00000010: 6af3 ba00 134d 7a76 3b25 21fe 4012 c002  j....Mzv;%!.@...
00000020: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000030: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000040: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000050: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000060: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000070: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000080: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000090: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000a0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000b0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000c0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000d0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000e0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
000000f0: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000100: 1100 0a00 0201 0010 0000 7016 1300 1300  ..........p.....
00000110: 1c00 5635 2e32 2e32 00ff ffff ffff ffff  ..V5.2.2........
00000120: ffff ffff ffff ffff ffff ffff ffff 1200  ................
00000130: 1c00 0200 0000 0010 0000 00f0 0800 0030  ...............0
00000140: 0100 0000 0900 0040 1500 0020 0e00 1400  .......@... ....
00000150: 4400 0200 0000 494d ad6c 716d f400 ad12  D.....IM.lqm....
00000160: 9a6d 7d21 74d8 ecfd f36d 3889 37a4 b58f  .m}!t....m8.7...
```

Now we know how to unpack the blob!

LZMA
(and potentially decryption)

https://github.com/ramikg/airoha-firmware-parser

Following the Command ID: Write Flash

```
switch(cmd_id) {
  case 0x400:
    pvVar3 = FUN_0838bad4(&inputMsg->hdr,length,transport_id);
    return pvVar3;
  case 0x402:
    pvVar3 = cmd_write_flash(&inputMsg->hdr,length,transport_id);
    return pvVar3;
  case 0x403:
    pvVar3 = FUN_0838b920(&inputMsg->hdr,length,transport_id);
    return pvVar3;
  case 0x404:
    pvVar3 = FUN_0838b994(&inputMsg->hdr,length,transport_id);
    return pvVar3;
}
```

Following the Command ID: Read Flash

```
switch(cmd_id) {
  case 0x400:
    pvVar3 = FUN_0838bad4(&inputMsg->hdr,length,transport_id);
    return pvVar3;
  case 0x402:
    pvVar3 = cmd_write_flash(&inputMsg->hdr,length,transport_id);
    return pvVar3;
  case 0x403:
    pvVar3 = cmd_read_flash(&inputMsg->hdr,length,transport_id);
    return pvVar3;
  case 0x404:
    pvVar3 = FUN_0838b994(&inputMsg->hdr,length,transport_id);
    return pvVar3;
}
```

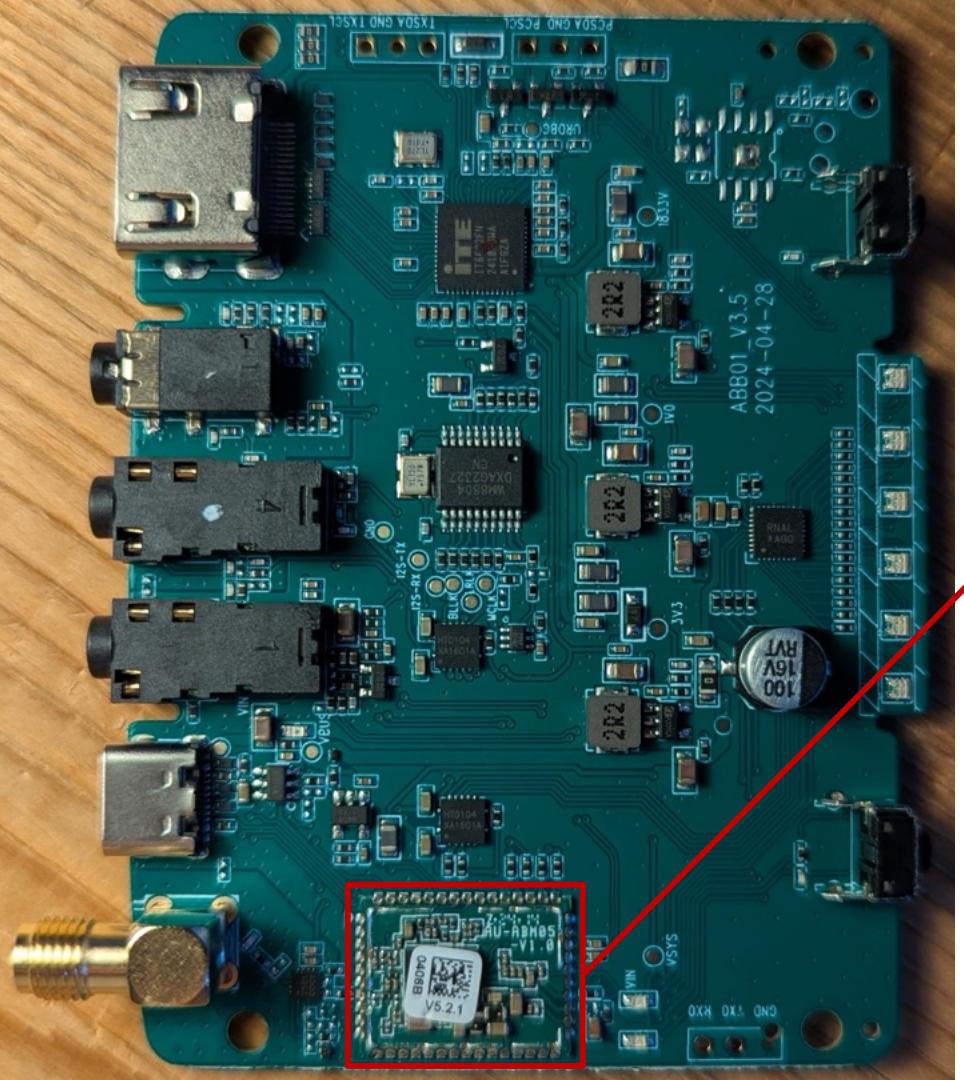
Found the Flash Read Command

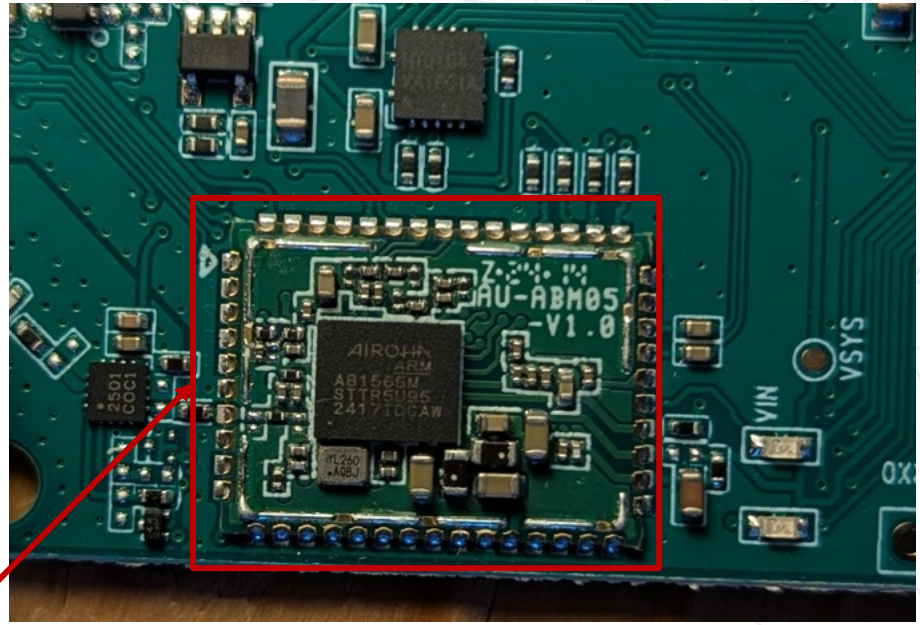
- We built a tool and implemented the protocol.
- We were able to dump flash via USB HID.
- And it worked for the headphones too!

At this point we were hooked and kind of forgot about our initial goal, the Auracast research.

Does that work with other, more prestigious devices?

Like these $9.000 Bose diamond crusted earbuds?





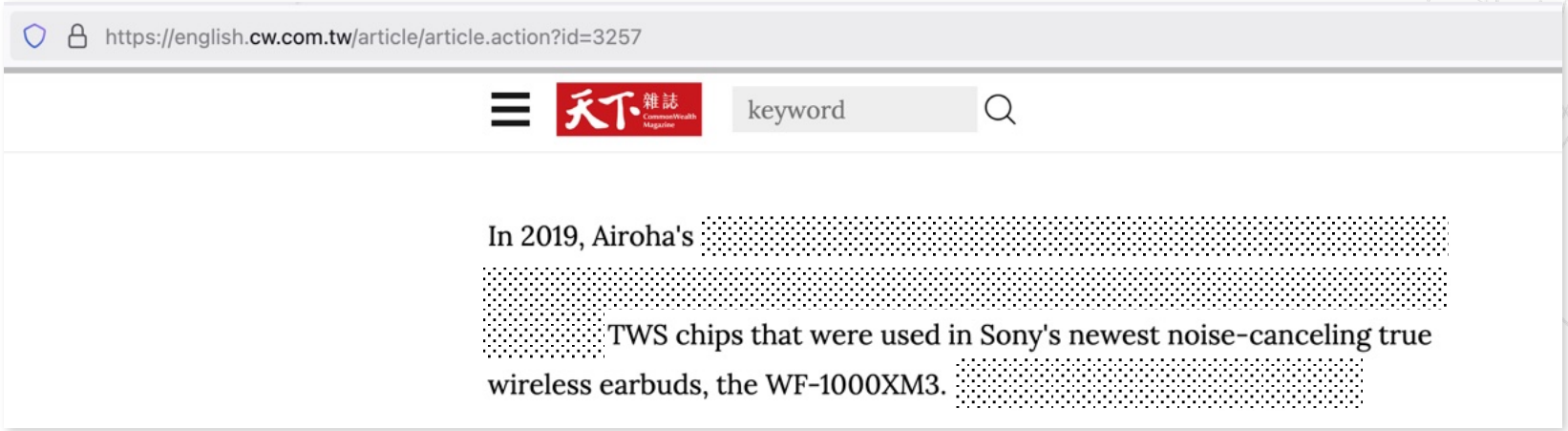
What do the headphones and transmitter have in common?

AIROHA

達發科技股份有限公司

AB1565M

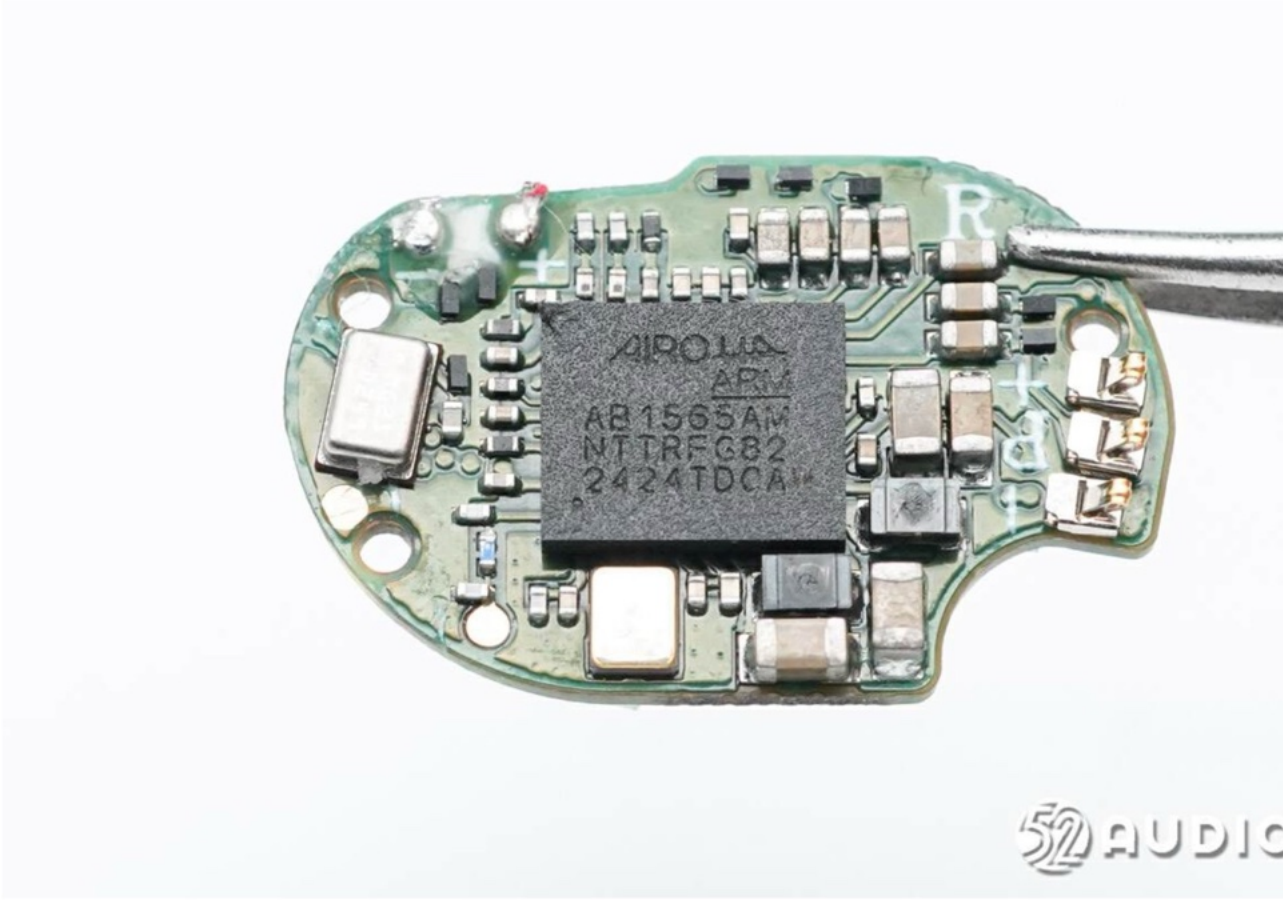
Search the internet



52audio.com

For extremely detailed teardowns and awesome pictures!

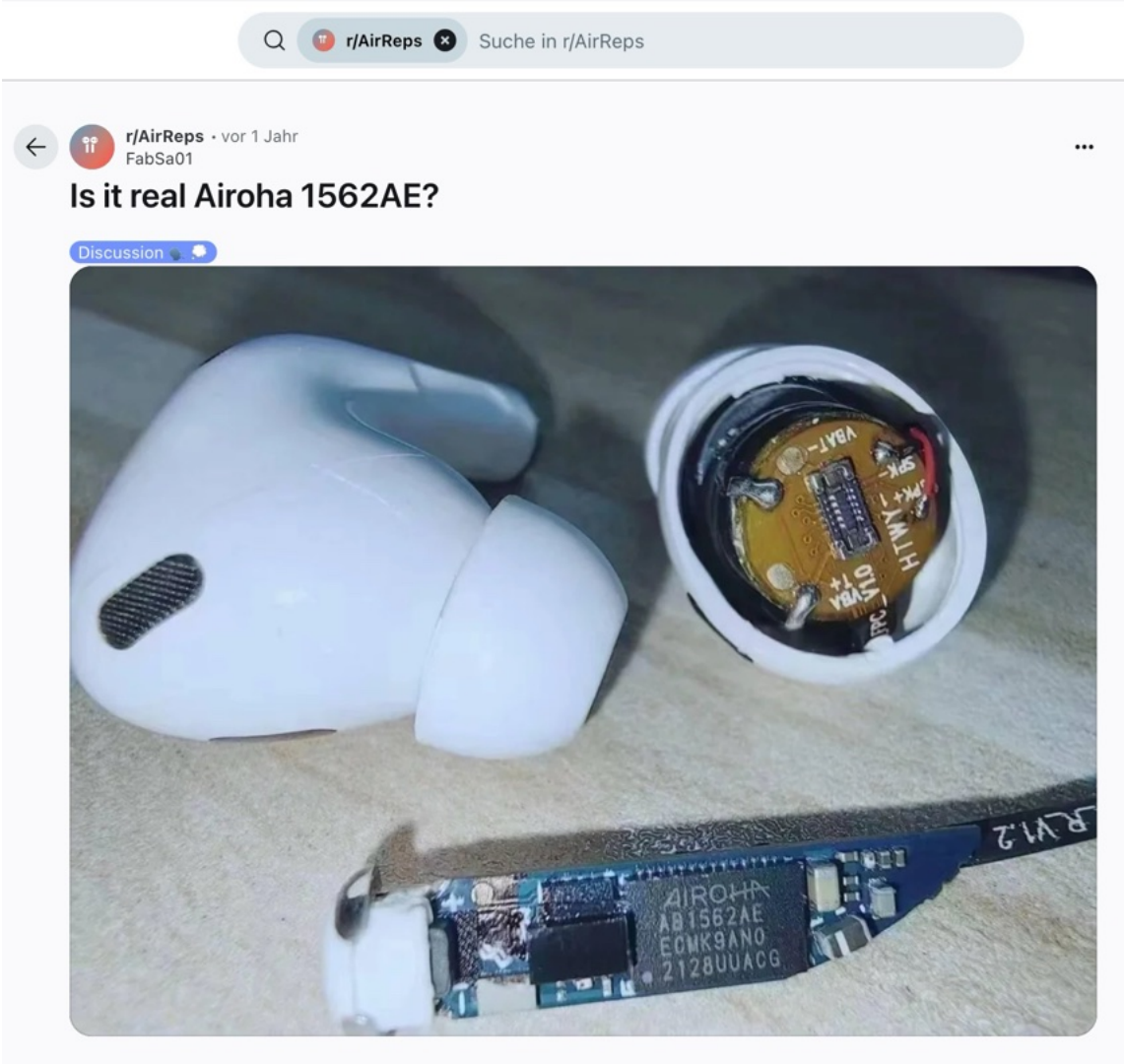
连接通话麦克风小板的金属弹片特写。



AIROHA达发（络达）AB1565AM蓝牙音频SoC，AB1565x系列是Airoha蓝牙进阶款音讯解决方案，支持蓝牙5.3，支持LE aud

AirPods replicas

There is an active community around Airoha-based AirPods replicas.



In fact, there are plenty of fancy devices with Airoha Bluetooth chips.

For example:	
Sony	JBL
Marshall	Beyerdynamic
Jabra	Bose

So, we bought some.

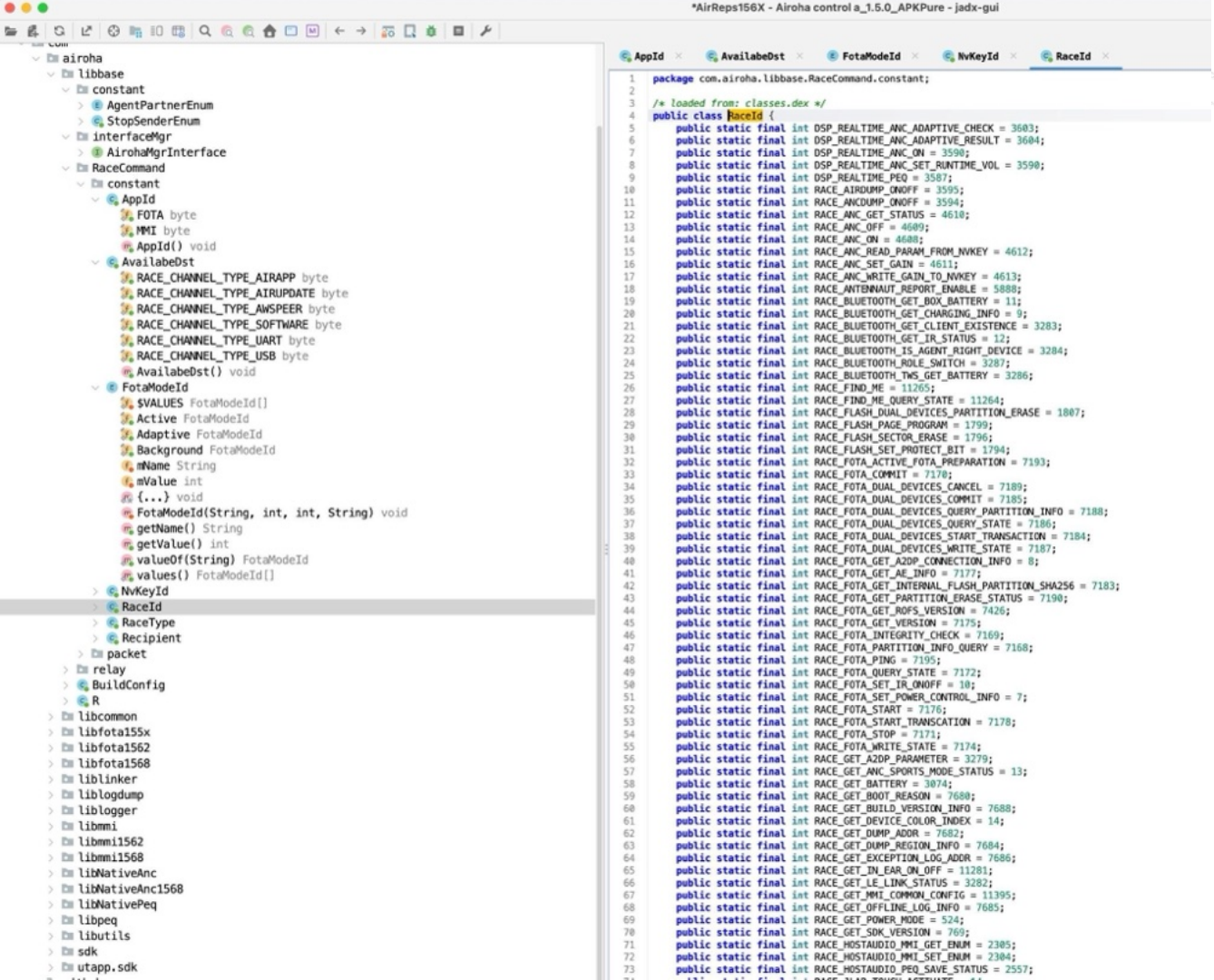
Some of our test device arsenal. Stay tuned for our headphone review YouTube channel!

No USB 😭

```
(.venv) frieder@fswork-ng ~/ERNW/research/le_audio/airoha sudo python hid_dump.py
Traceback (most recent call last):
  File "/home/frieder/ERNW/research/le_audio/airoha/hid_dump.py", line 158, in <module>
    device = find_hid_device(VID, PID)
  File "/home/frieder/ERNW/research/le_audio/airoha/hid_dump.py", line 13, in find_hid_device
    return hid.Device(vid, pid)
           ~~~~~~~~~~^^^^^^^^^^
  File "/home/frieder/ERNW/research/le_audio/.venv/lib/python3.13/site-packages/hid/__init__.py
", line 174, in __init__
    raise HIDException('unable to open device')
hid.HIDException: unable to open device
```

But how do you update the headphones?

There‘s an app for that!



Sony SoundConnect

AirReps 156X App

...

The protocol is called:

RACE

Quick Interlude: Bluetooth Classic vs. BLE

Bluetooth Classic	Bluetooth Low Energy
Built for high data rate (e.g. audio)	Built for energy efficiency
Public device address	Public/Random addresses for privacy
Discoverable vs. Non-discoverable	Advertisements
RFCOMM/L2CAP	GATT

Bluetooth Classic as Transport

- **RACE** uses the RFCOMM profile as transport.
- **RFCOMM** is a Bluetooth Classic protocol, emulates serial interface
- **RFCOMM** has channels that are identified by their channel number
- **Channel numbers** can be discovered using the **SDP** protocol and a channel UUID.
- **Channel UUIDs** for RACE are vendor specific.

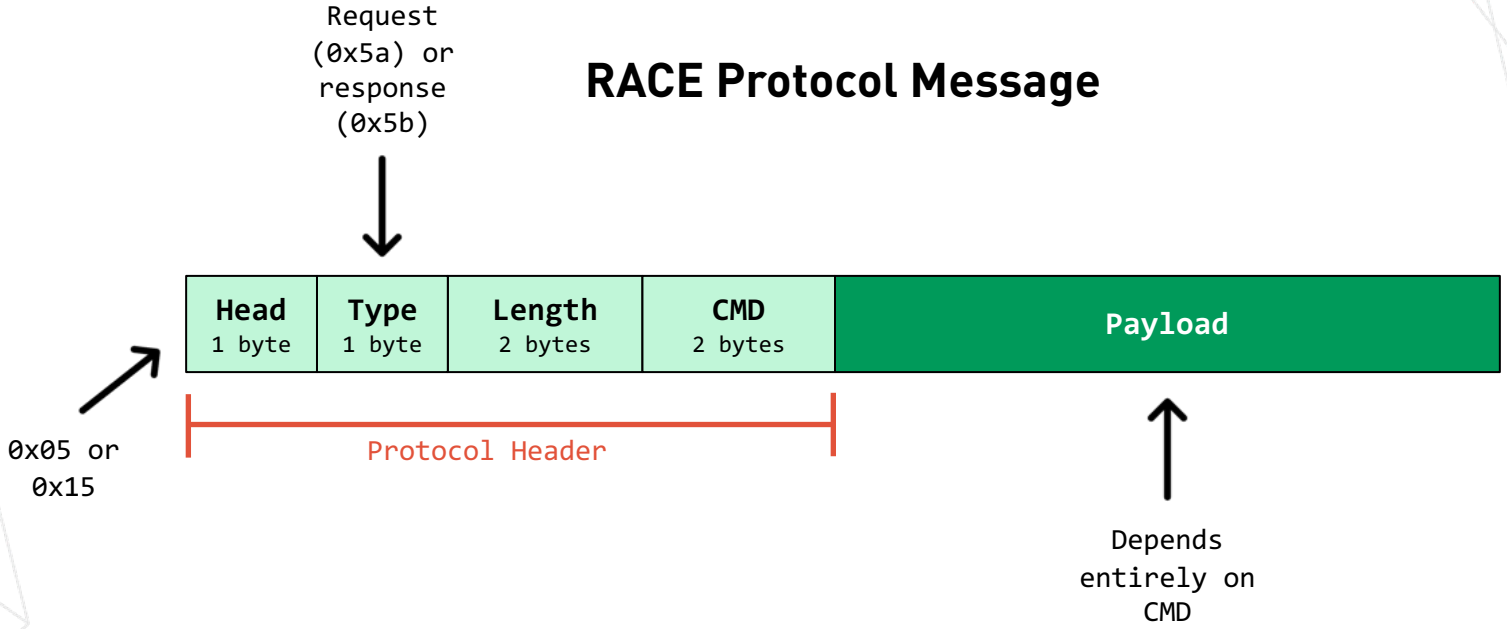
```
COMMON_SPP_UUID =    UUID("00001101-0000-1000-8000-00805F9B34FB")
AIROHA_SPP_UUID =    UUID("00000000-0000-0000-0099-AABBCCDDEEFF")
SONY_SPP_UUID =      UUID("8901DFA8-5C7E-4D8F-9F0C-C2B70683F5F0")
BOSE_SPP_UUID =      UUID("2D064AA9-32B5-4970-865C-643742BD2862")
BEYER_SPP_UUID =     UUID("7B46E8DE-5A7E-4512-BE0D-863AEAA3312B")
```

Bluetooth Low Energy as Transport

- **RACE** is also available via GATT.
- **GATT** has services.
- **Services** have characteristics.
- **Characteristics** can be read from, written and subscribed to.
- **Services & Characteristics** can be discovered and have UUIDs.
- **UUIDs** for RACE are sometimes vendor specific.

```
AIROHA_GATT_SERVICE_UUID =   UUID("5052494D-2DAB-0341-6972-6F6861424C45")
AIROHA_GATT_TX_UUID =        UUID("43484152-2DAB-3241-6972-6F6861424C45")
AIROHA_GATT_RX_UUID =        UUID("43484152-2DAB-3141-6972-6F6861424C45")
SONY_GATT_SERVICE_UUID =     UUID("dc405470-a351-4a59-97d8-2e2e3b207fbb")
SONY_GATT_TX_UUID =          UUID("bfd869fa-a3f2-4c2f-bcff-3eb1ec80cead")
SONY_GATT_RX_UUID =          UUID("2a6b6575-faf6-418c-923f-ccd63a56d955")
```

RACE Protocol Message



RACE: Read Flash

Head 1 byte	Type 1 byte	Length 2 bytes	CMD 2 bytes	Payload		
0x05	0x5a	0x0008	0x0403	0x00	0x01	0x08000000
				Type 1 byte	Num 1 byte	Address 4 bytes

RACE: Get Build Version

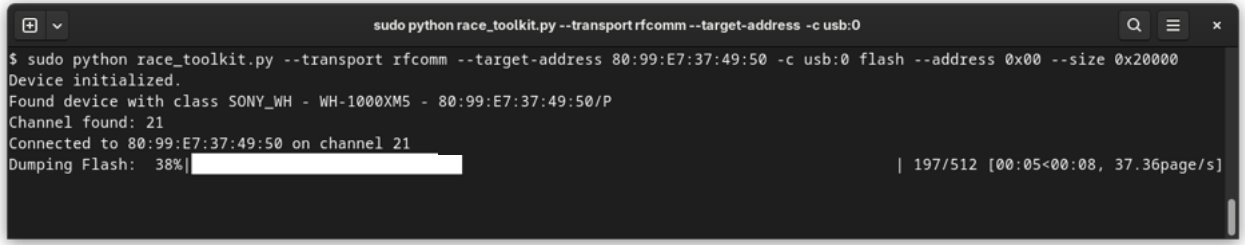
Head 1 byte	Type 1 byte	Length 2 bytes	CMD 2 bytes	Payload
0x05	0x5a	0x0008	0x1E08	-

Things we know RACE can do:

- Read (and write*) flash.
- Read and write RAM/register/peripheral (bytewise from any address).
- FOTA (Firmware Over the Air) operations.
- Get Bluetooth Classic device address.
- Read out build version.
- … many other commands!

- Not every device supports all these functions.

Tool Time



- We built a tool that implements the RACE protocol via RFCOMM and GATT.
- Using Google's Python-based Bluetooth library Bumble.
- This is nice for research on our headphones: Dumping flash, debugging via RAM read/write.

But then we realized something...

Umm...

- During testing we always put our devices in pairing mode before connecting.
- Turns out: pairing is not required. No authentication.
- Anyone in Bluetooth range can connect to our headphones!
- Anyone in Bluetooth range can dump our headphones' flash.
- Anyone in Bluetooth range can read & write our headphones' RAM.

Live Demos!

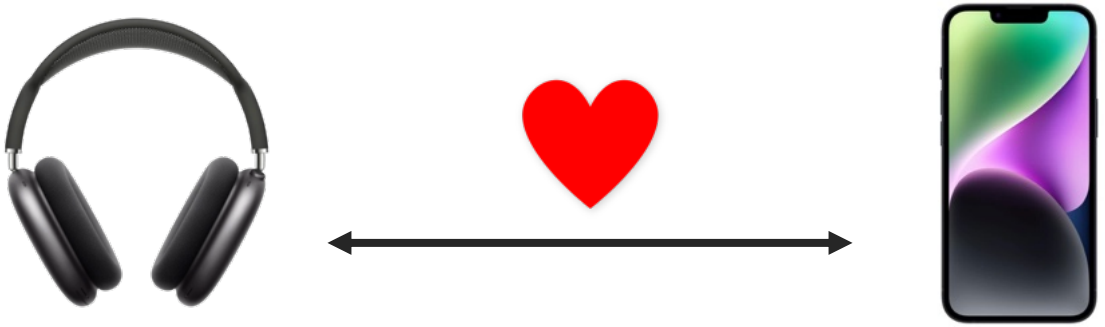
Extract data

Eavesdrop

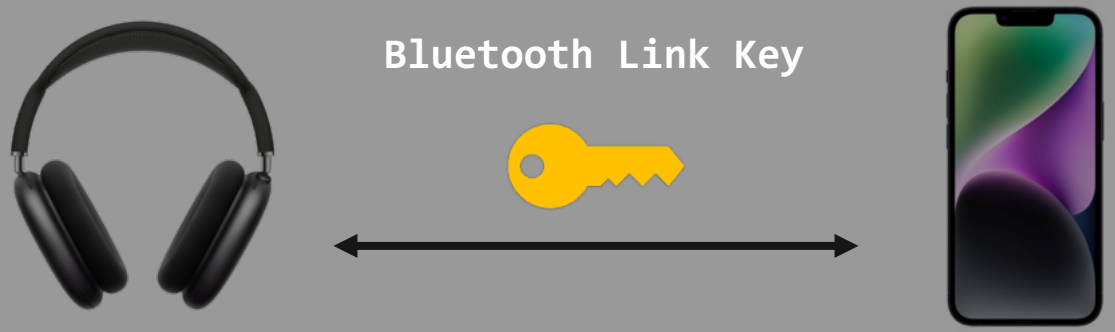
Inject Code

Pivoting

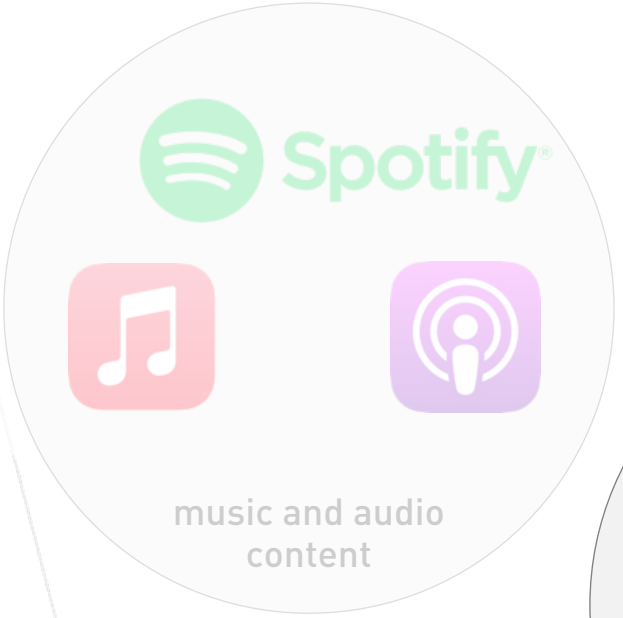
At the center there's this love relationship

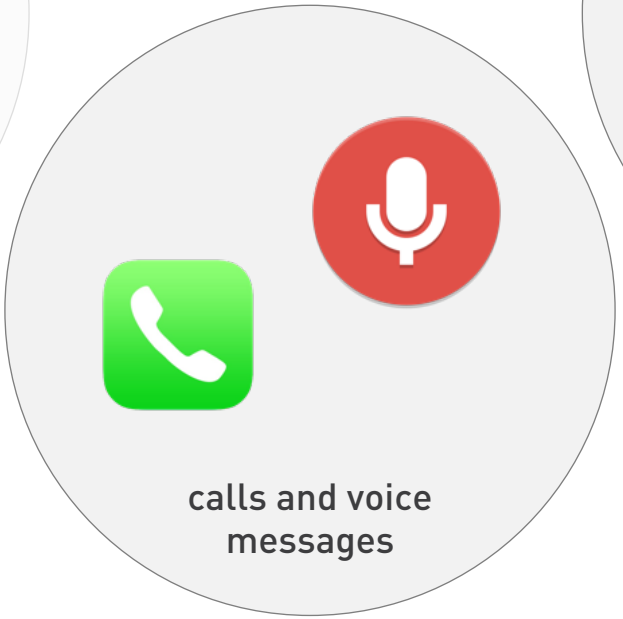


Or rather:
This cryptographic bond



What do people use headphones for?



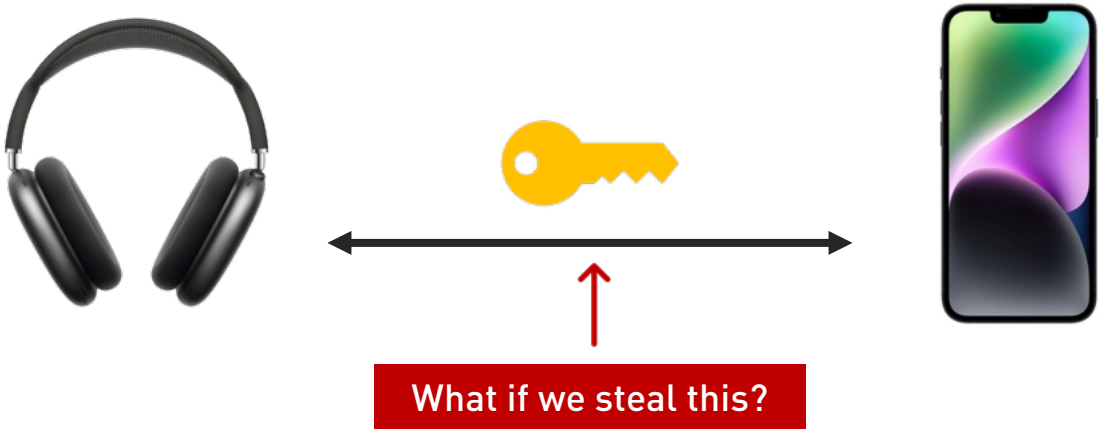


Hands-Free Profile (HfP)

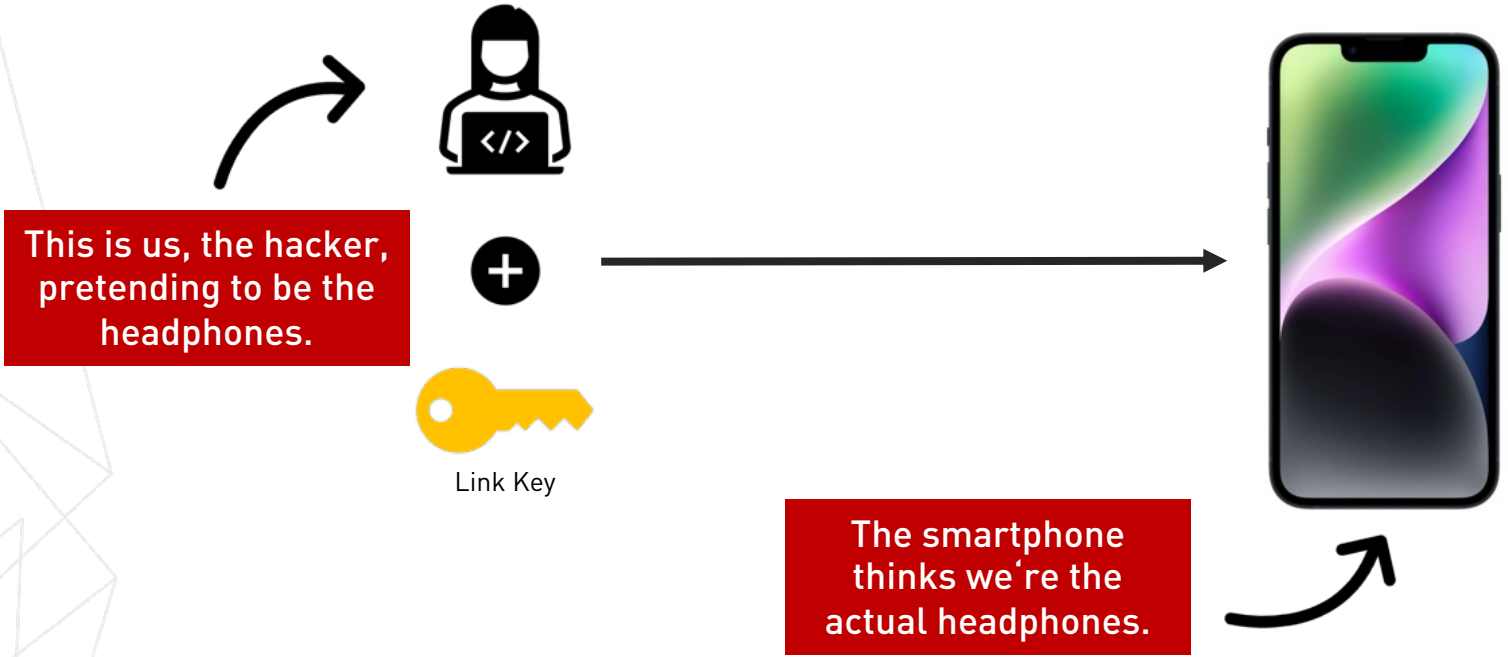
Bluetooth Classic: Hands-Free Profile (HFP)

- **Call Control**
  - Answer, reject, end calls
  - Dial a number (including redialing the last number)
- **Audio Routing**
  - Transfer audio between phone and hands-free device
- **Caller Information Access**
  - Read caller ID
  - Access signal strength, battery level
- **Network Information**
  - Query carrier/operator name
  - Subscriber Number (local phone number)
- **Voice Assistant**
  - Trigger Voice Assistant (Gemini/Google Assistant/Siri)
- **Phone Book Access**
  - Non-standard AT commands for phonebook access
  - Requires permissions

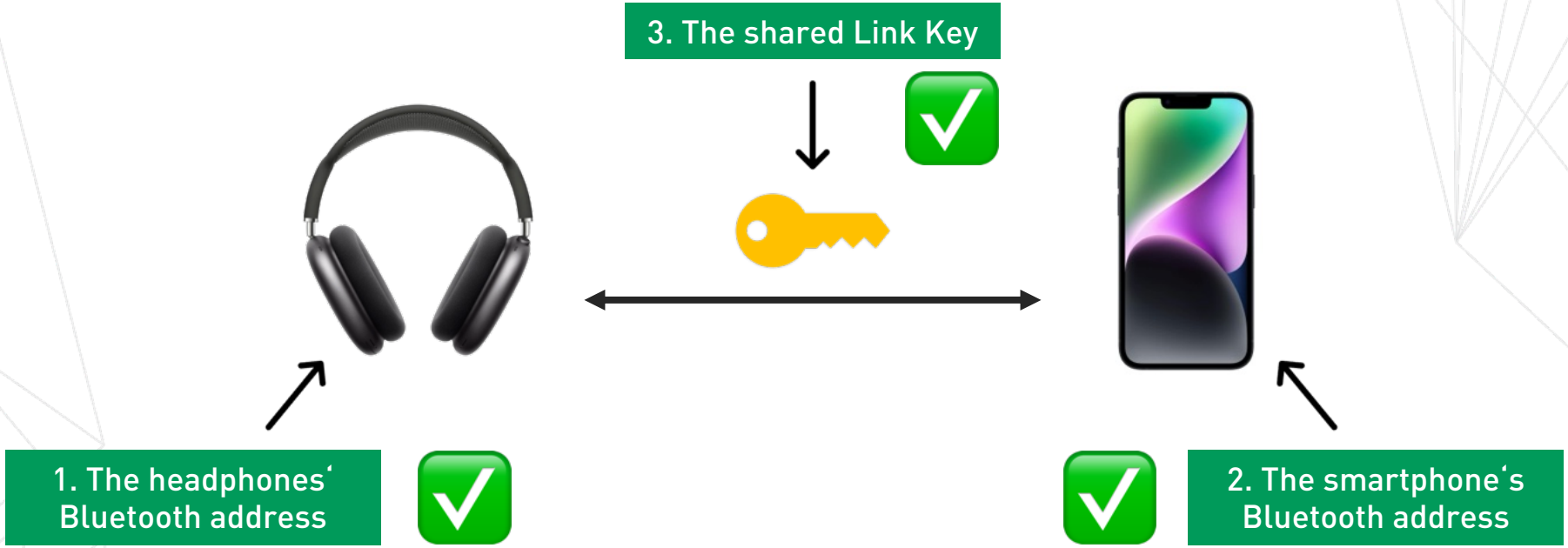
Let's get back to the cryptographic bond



Impersonating the Headphones to the Smartphone



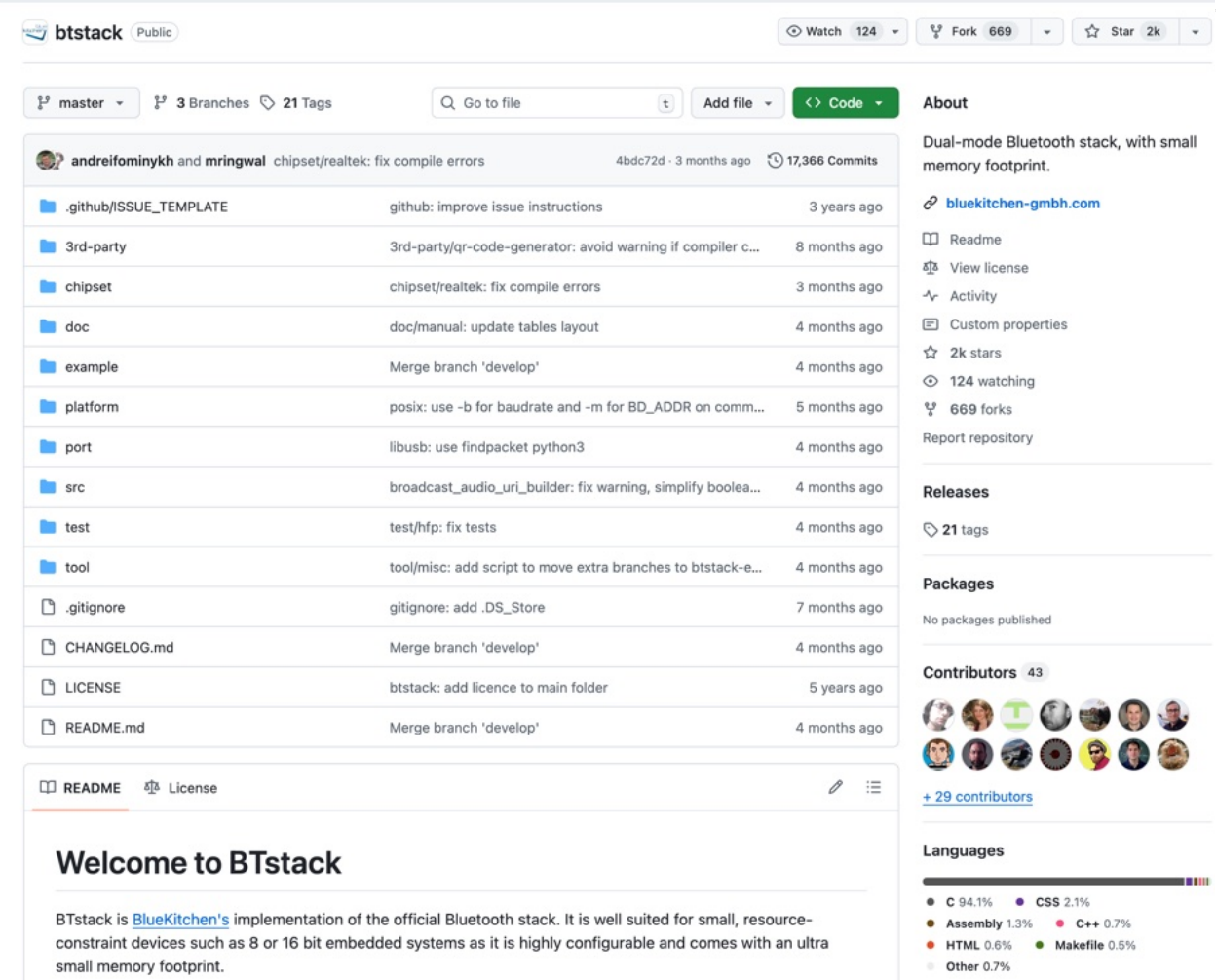
But what do we need for that?



Live Demo


WLAN Call
Sun 21. Dec
10:40
Amazon
2m ago
Amazon: 624348 is your sign-in code. For your security, don't share it. Amazon will never contact you to ask for this code.
WhatsApp
Show less
Logged out of WhatsApp
3m ago
Your phone number is no longer registered on this phone. Tap to view.
WhatsApp
4m ago
WhatsApp registration code was requested for your phone number.
Code for new phone
5m ago
Do not share it with anyone. Your code is 420592.

https://github.com/bluekitchen/btstack



btstack Public

Watch 124 | Fork 669 | Star 2k

master | 3 Branches | 21 Tags | Go to file t | Add file | Code

andreifominykh and mringwal chipset/realtek: fix compile errors 4bdc72d · 3 months ago 17,366 Commits

.github/ISSUE_TEMPLATE	github: improve issue instructions	3 years ago
3rd-party	3rd-party/qr-code-generator: avoid warning if compiler c...	8 months ago
chipset	chipset/realtek: fix compile errors	3 months ago
doc	doc/manual: update tables layout	4 months ago
example	Merge branch 'develop'	4 months ago
platform	posix: use -b for baudrate and -m for BD_ADDR on comm...	5 months ago
port	libusb: use findpacket python3	4 months ago
src	broadcast_audio_uri_builder: fix warning, simplify boolea...	4 months ago
test	test/hfp: fix tests	4 months ago
tool	tool/misc: add script to move extra branches to btstack-e...	4 months ago
.gitignore	gitignore: add .DS_Store	7 months ago
CHANGELOG.md	Merge branch 'develop'	4 months ago
LICENSE	btstack: add licence to main folder	5 years ago
README.md	Merge branch 'develop'	4 months ago

README | License

Welcome to BTstack

BTstack is BlueKitchen's implementation of the official Bluetooth stack. It is well suited for small, resource-constraint devices such as 8 or 16 bit embedded systems as it is highly configurable and comes with an ultra small memory footprint.

About

Dual-mode Bluetooth stack, with small memory footprint.

bluekitchen-gmbh.com

Readme
View license
Activity
Custom properties
2k stars
124 watching
669 forks
Report repository

Releases

21 tags

Packages

No packages published

Contributors 43

+ 29 contributors

Languages

C 94.1% CSS 2.1%
Assembly 1.3% C++ 0.7%
HTML 0.6% Makefile 0.5%
Other 0.7%

Shoutout to BTStack!

Disclosure

These are the devices we initially saw and were able to confirm

Bose
- Quiet Comfort Earbuds[1,4]

MoerLabs
- EchoBeatz

Jabra
- Elite 8 Active[1,3]

Jlab
- Epic Air Sport ANC

JBL
- Live Buds 3
- Endurance Race 2

Beyerdynamic
- Amiron 300

Teufel
- Airy TWS 2

Sony
- WH-1000XM6[2]
- WF-1000XM3
- WF-1000XM5
- CH-720N
- WH-1000XM5[2,3]
- WF-1000XM4
- WH-1000XM4[2,3]
- WH-XB910N
- WI-C100
- WH-CH520
- ULT Wear
- WF-C510-GFP
- WF-C500
- Link Buds S

Marshall
- WOBURN III
- STANMORE III
- ACTON III
- MAJOR V
- MINOR IV
- MOTIF II

Things we noticed (not exhaustive):
1: No BLE (GATT)
2: Some protocol messages are filtered
3: Classic pairing is required
4: Classic pairing is sometimes(?) required

This is our current *assumption*

Bose
- Quiet Comfort Earbuds

MoerLabs
- EchoBeatz

Jabra
- Elite 8 Active

Jlab
- Epic Air Sport ANC

JBL
- Live Buds 3
- Endurance Race 2

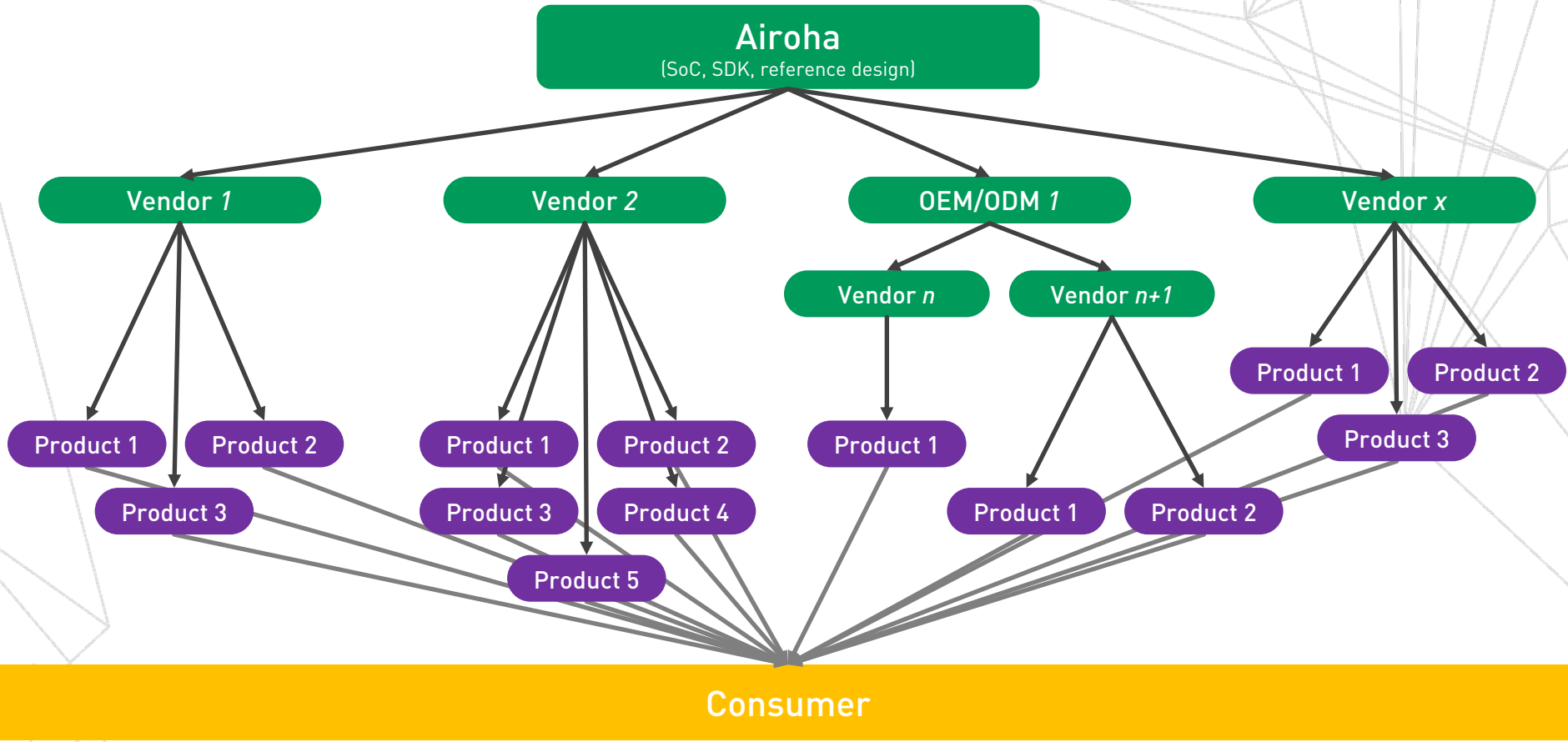
Beyerdynamic
- Amiron 300

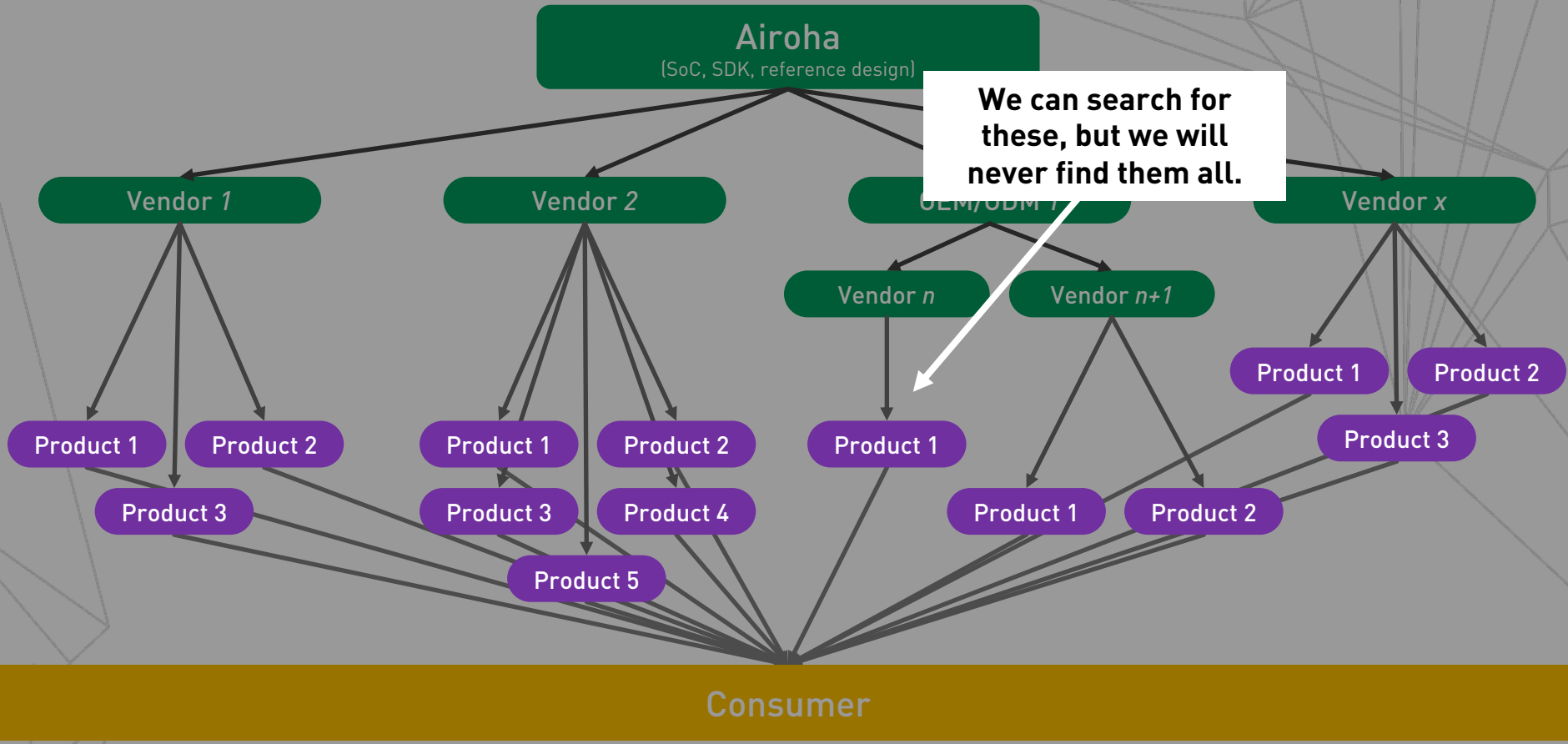
Teufel
- Airy TWS 2

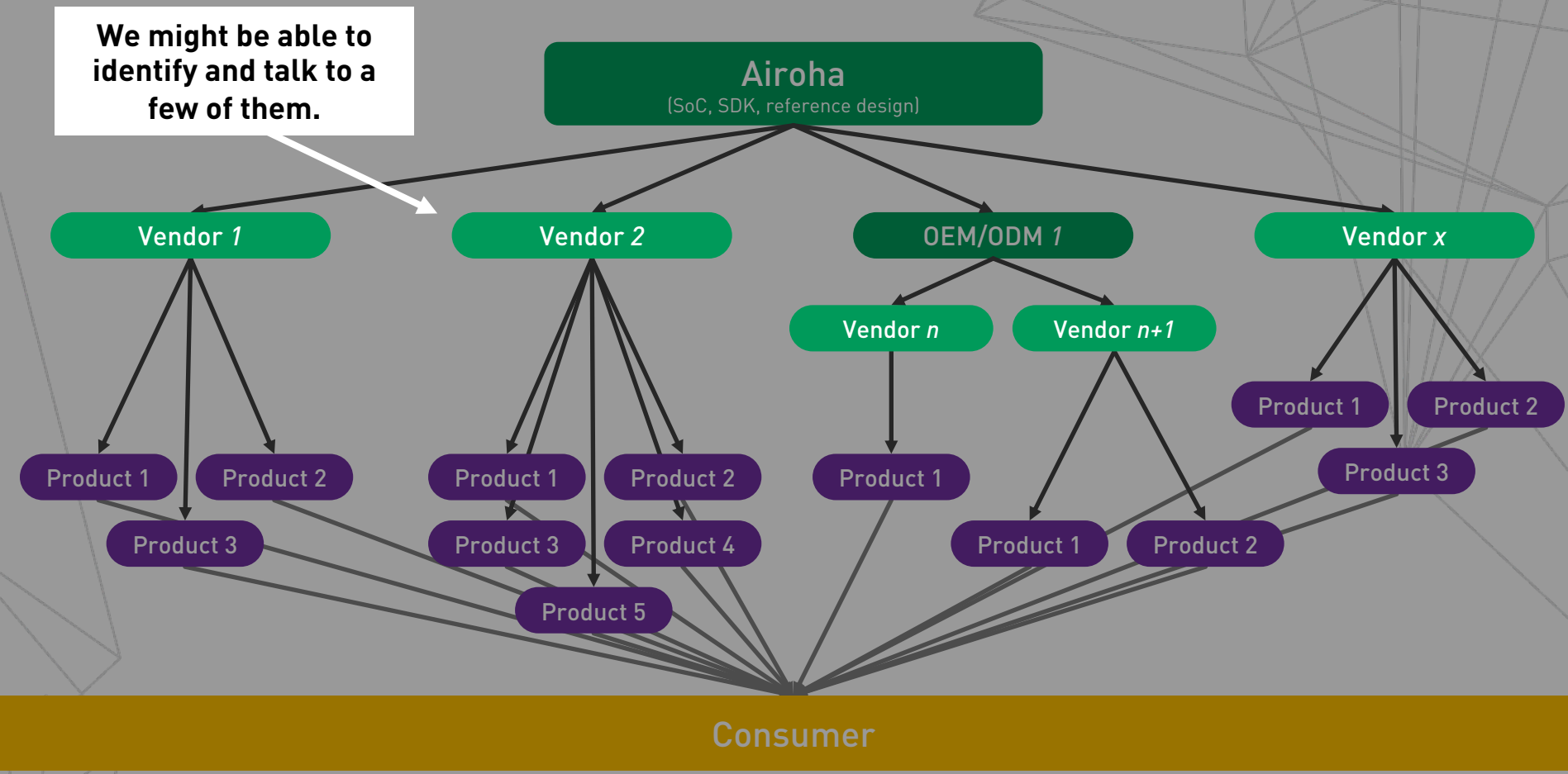
Sony
- WH-1000XM6
- WF-1000XM3
- WF-1000XM5
- CH-720N
- WH-1000XM5
- WF-1000XM4
- WH-1000XM4
- WH-XB910N
- WI-C100
- WH-CH520
- ULT Wear
- WF-C510-GFP
- WF-C500
- Link Buds S

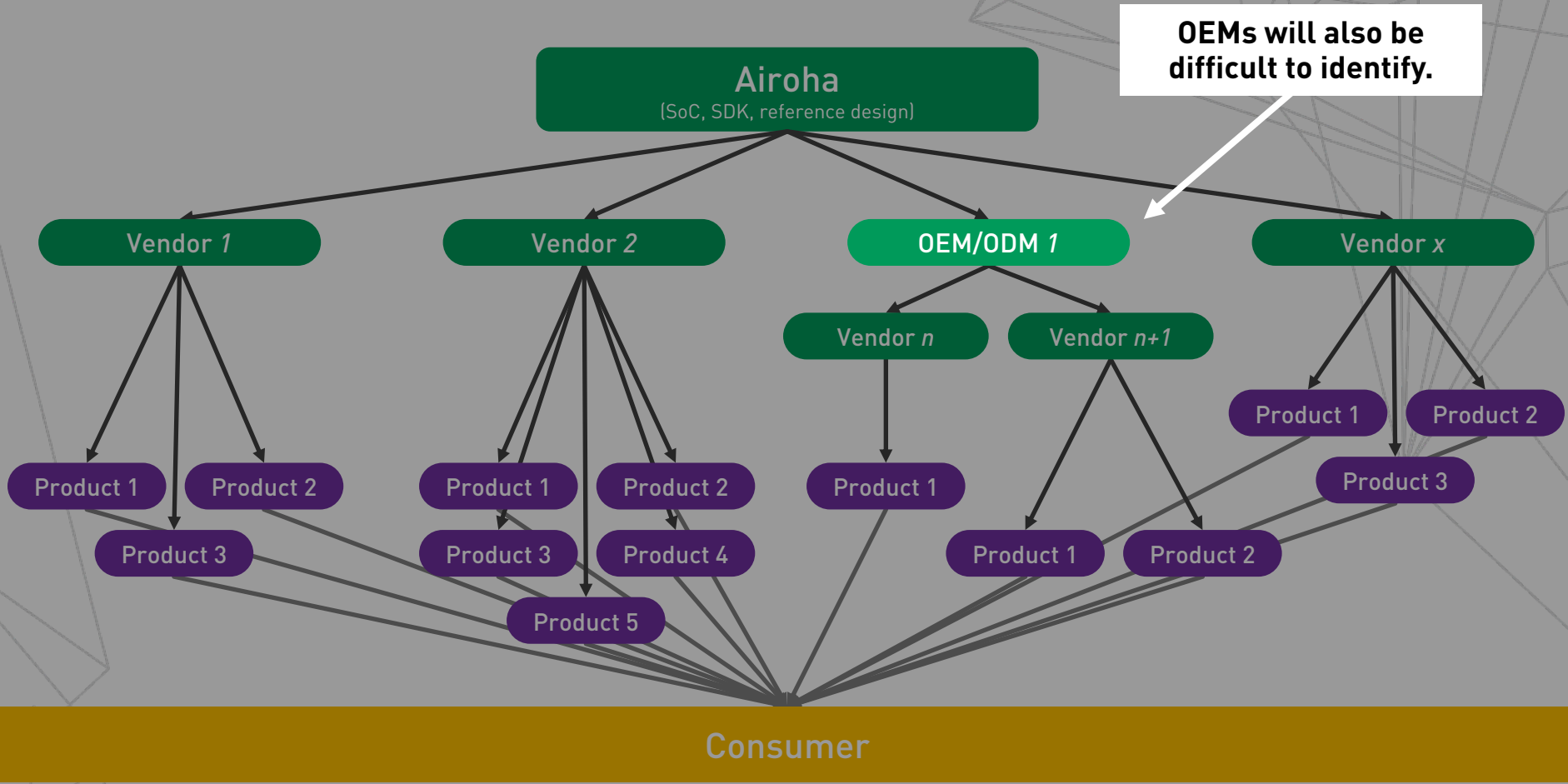
Marshall
- WOBURN III
- STANMORE III
- ACTON III
- MAJOR V
- MINOR IV
- MOTIF II

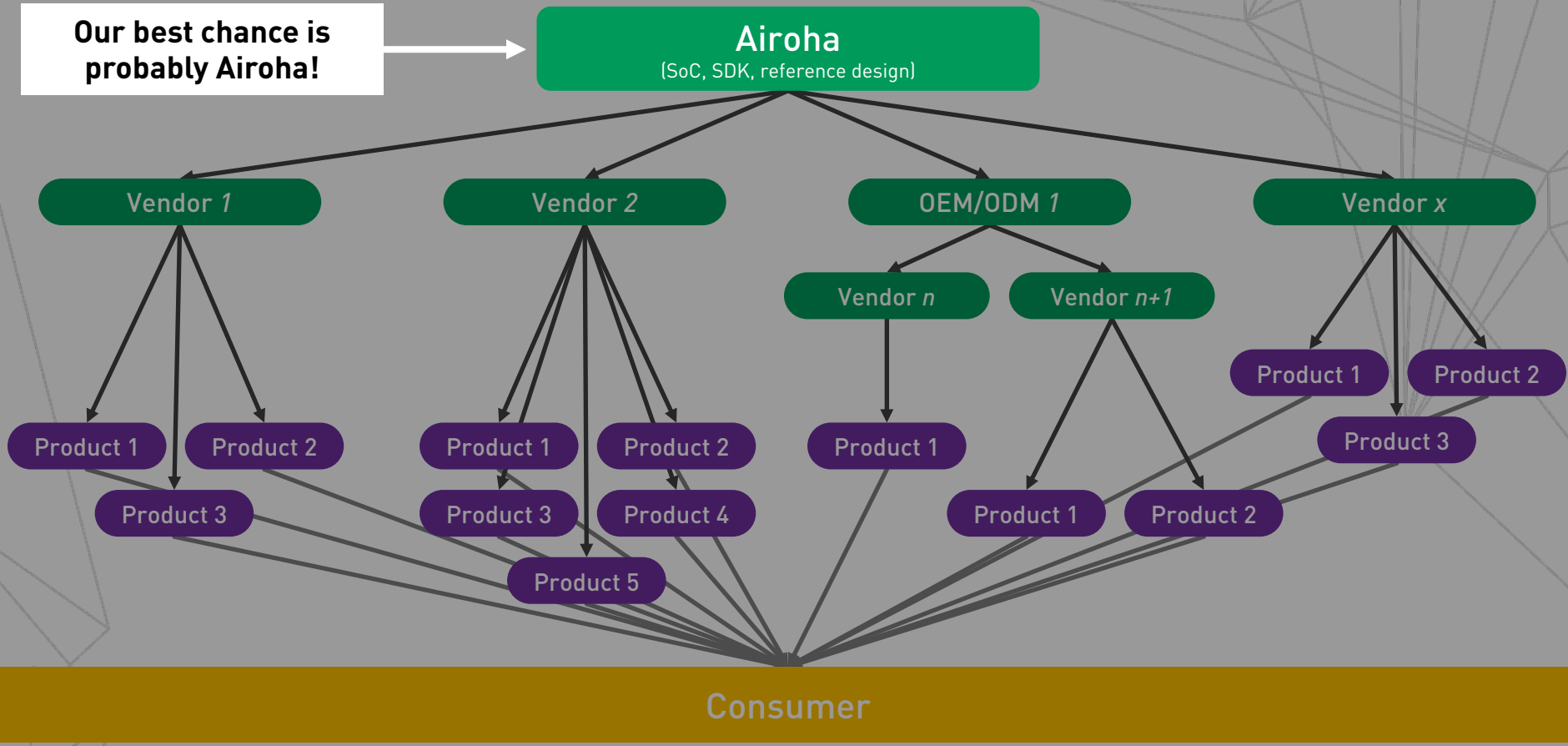
- Some vendors have released updates since the disclosure.
- Only some of them explicitly mentioned fixing the CVEs.
- Some put: “Fix a security vulnerability” or “Stability and performance improvement”
- We might be missing patches! This is only *our* current overview.
- We don’t have the time and resources to check and monitor updates.


Airoha
(SoC, SDK, reference design)
Vendor 1
Vendor 2
OEM/ODM 1
Vendor x
Vendor n
Vendor n+1
Product 1
Product 2
Product 3
Product 1
Product 2
Product 3
Product 4
Product 5
Product 1
Product 1
Product 2
Product 1
Product 2
Product 3
Consumer

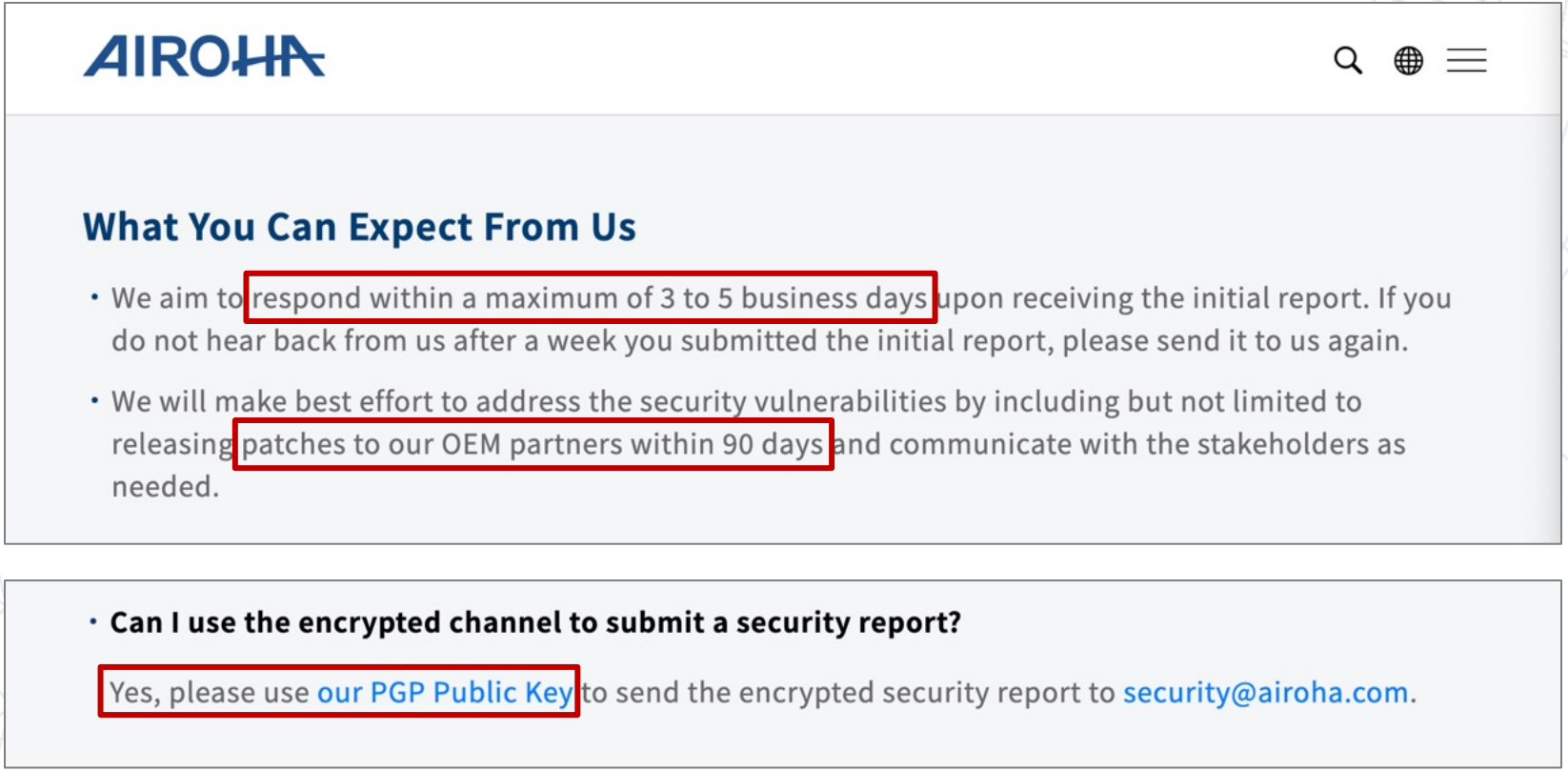

Airoha
(SoC, SDK, reference design)
We can search for these, but we will never find them all.
Vendor 1
Vendor 2
Vendor n
Vendor n+1
Vendor x
Product 1
Product 2
Product 3
Product 4
Product 5
Consumer


We might be able to identify and talk to a few of them.
Airoha
(SoC, SDK, reference design)
Vendor 1
Vendor 2
OEM/ODM 1
Vendor x
Vendor n
Vendor n+1
Product 1
Product 2
Product 3
Product 4
Product 5
Consumer


OEMs will also be difficult to identify.
Airoha
(SoC, SDK, reference design)
Vendor 1
Vendor 2
OEM/ODM 1
Vendor x
Vendor n
Vendor n+1
Product 1
Product 2
Product 3
Product 4
Product 5
Consumer


Our best chance is probably Airoha!
Airoha
(SoC, SDK, reference design)
Vendor 1
Vendor 2
OEM/ODM 1
Vendor x
Vendor n
Vendor n+1
Product 1
Product 2
Product 3
Product 4
Product 5
Consumer

Airoha's security disclosure page looks really good!



Disclosure Timeline: Short Version

March 25th

ERNW contacts Airoha with technical writeup of vulnerabilities and list of currently known vulnerable devices.

April 24th

After several different communication attempts ERNW tells Airoha it will contact some of the vendors about the issue.

ERNW contacts three vendors (Sony, Marshall, Beyerdynamic) of whom only one responds (Beyerdynamic).

May 27th

Airoha's first response to ERNW, due to technical issues on their side. Communication and exchange between ERNW starts. Airoha is committed to fix issues with priority. Communication with Airoha is very positive.

June 4th

Airoha supplies device manufacturers with an SDK update with mitigations. Manufacturers start developing and shipping patches for their products.

Disclosure Timeline: Short Version

June 26th

93 days after initial report. We partially disclose the vulnerabilities at TROOPERS and with security advisory post.

Because patching is difficult and takes time, we agree to postpone full disclosure. Most devices were still vulnerable at this point.

June - December

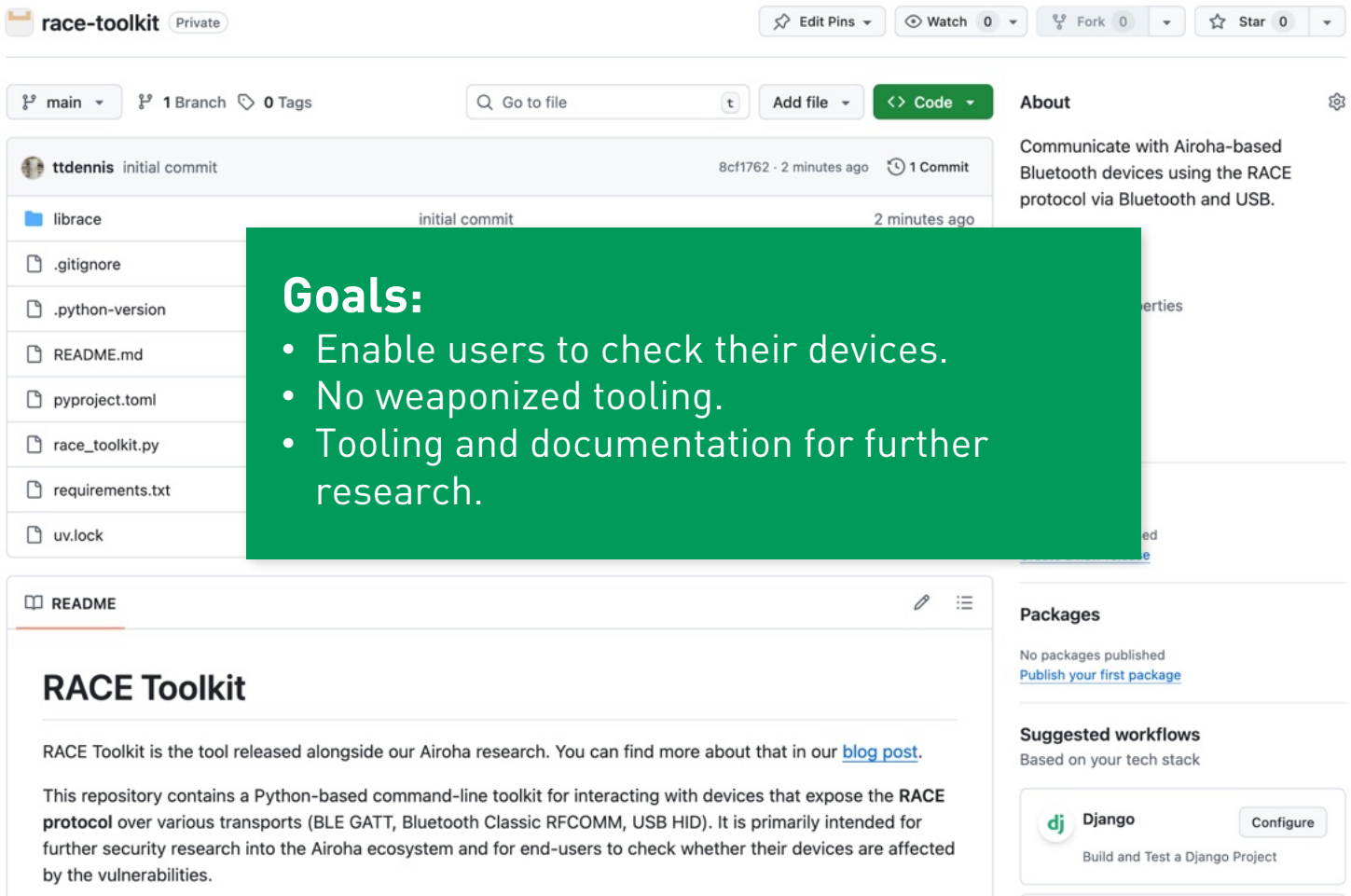
Many vendors release updates, but only some of them explicitly mention the vulnerabilities (Jabra, Marshall, Beyerdynamic).

We don't have time or resources to monitor and check them all. To us, the situation is intransparent.

End of Year - 39c3

Full disclosure of technical details. Release of tooling and documentation.

https://github.com/auracast-research/race-toolkit



FOTA – Firmware Updates

- We reimplemented the firmware update mechanism.
  - flash valid, "signed" firmware files
  - Sony: MDR Proxy repository (https://github.com/lzghzr/MDR_Proxy)
- It's also possible to downgrade firmware.
  - Only for research, for your personal device want to have the newest FW with the vulns patched.
- Unpack and pack the firmware to potentially manipulate it
  - ❗ during that process we bricked two devices, so we currently don't recommend trying that.
- Currently only works for headphones, not for True Wireless earbuds.

6.1.1 nRF Connect Mobile on iOS

Step 1: Filtering

The first thing that should be done is setting a filter. There are usually many Bluetooth devices in the vicinity. Tapping the arrow button on the top right opens up the filter menu. Switch on *Remove Unnamed* and *Remove Unconnectable*. We just want to show devices that are actually connectable, and we want to be able to identify them by their name. If the target device was not renamed explicitly, it should be advertised under its model name (e.g. `WH-CH720N` in this example).

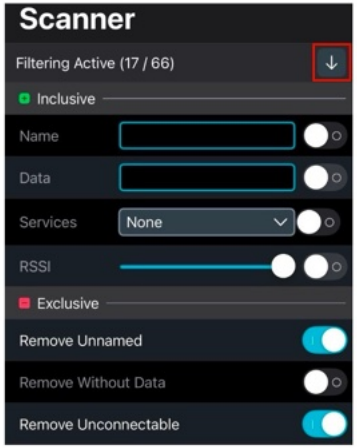


Figure 17: iOS Set Filter

Step 2: Connecting

Once the filter is active, you can look for the target device. In this example, we're connecting to the Sony WH-CH720N headphones. Tap the *Connect* button to establish a BLE connection. Some device models might request pairing at this point. Accept the pairing if necessary. If there's no prompt for a pin or button press, this is still not a proper authentication.

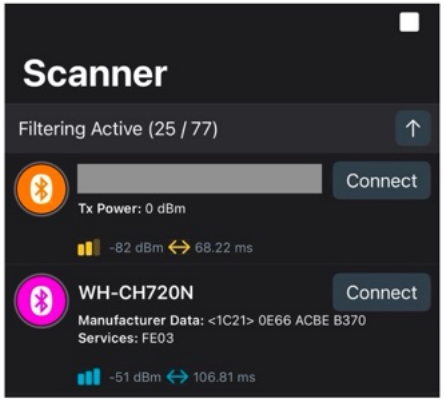


Figure 18: iOS Connect to Device

Blog Post/White Paper:
Check if your device is affected via BLE using the Nordic nRF Connect Mobile App. This is a low-barrier method to quickly check a device.

A quick note on mitigation

Mitigation

- Enforce authentication for Bluetooth Classic connections (CVE-2025-20701).
- Enforce authentication for sensitive GATT services (CVE-2025-20700).
- Don’t expose critical debug-like interfaces via Bluetooth (CVE-2025-20702).

Manufacturers

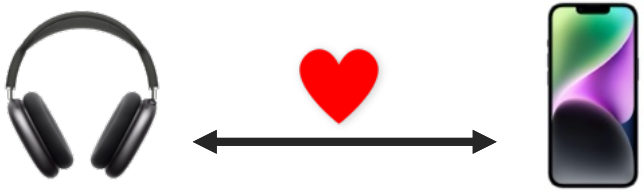
- Do a security assessment before you sell your devices.
- E.g. BSAM: Bluetooth security assessment methodology. This would have caught at least the pairing issue.
- Offer security contacts in addition to bug bounty platforms.

Users

- Update your headphones!
- If your vendor does not mention the CVEs, check your device with race-toolkit.
- Maybe use cable headphones if you feel like you might be a target.
- Remove old Bluetooth pairings.

Closing Thoughts

- As smartphones are getting more secure, peripherals become more interesting for attackers.
- Even physical compromise of peripherals may be worth the effort if it allows access to the phone.



Blog Post

GitHub: RACE Toolkit

- Check the blog post & white paper for more technical details and how to check your device.
- Come find us at 39c3 if you have questions about the tooling, your headphones or think you found another affected device!

- Links:
  - Blogpost: https://insinuator.net/2025/12/bluetooth-headphone-jacking-full-disclosure-of-airoha-race-vulnerabilities
  - RACE Toolkit: https://github.com/auracast-research/race-toolkit
  - Whitepaper: https://ernw.de/en/publications.html

dheinze at ernw.de
fsteinmetz at ernw.de

ttdennis@chaos.social
twillnix@infosec.exchange