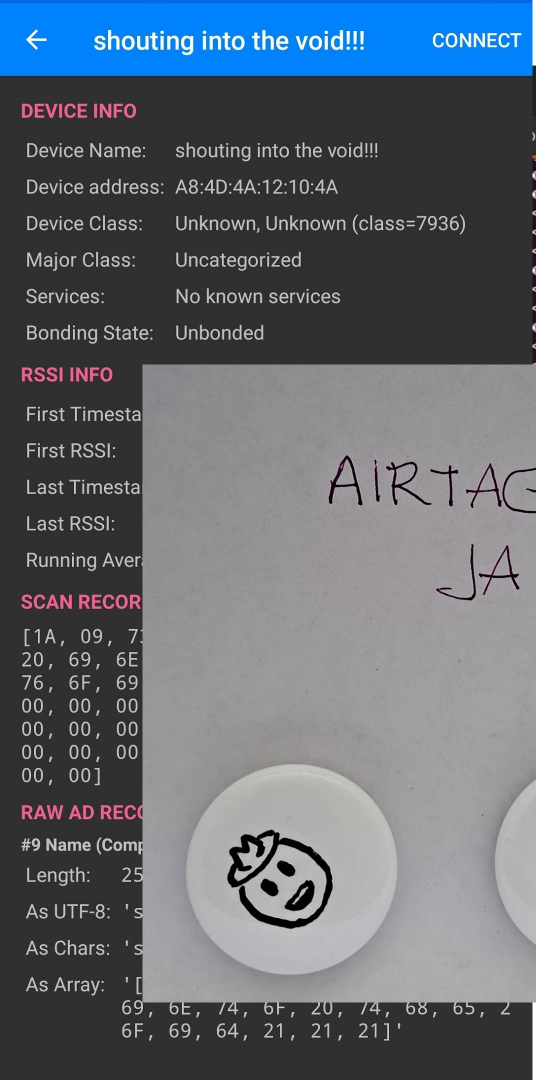

shouting into the void!!!

CONNECT

DEVICE INFO

```
Device Name:     shouting into the void!!!
Device address:  A8:4D:4A:12:10:4A
Device Class:    Unknown, Unknown (class=7936)
Major Class:     Uncategorized
Services:        No known services
Bonding State:   Unbonded
```

RSSI INFO

```
First Timesta
First RSSI:
Last Timesta
Last RSSI:
Running Aver
```

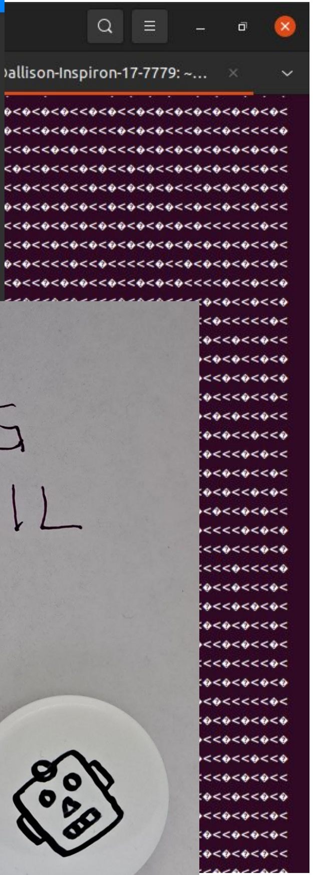
SCAN RECOR

```
[1A, 09, 7
20, 69, 6E
76, 6F, 69
00, 00, 00
00, 00, 00
00, 00, 00
00, 00]
```

RAW AD REC

#9 Name (Comp

```
Length:    25
As UTF-8:  's
As Chars:  's
As Array:  '[
           69, 6E, 74, 6F, 20, 74, 68, 65, 2
           6F, 69, 64, 21, 21, 21]'
```



Fantastic AirTags and Where (Not) to Find Them

Alex Bellon & Allison Turner

Stuff's Movin *Fast* Over Here, Y'all

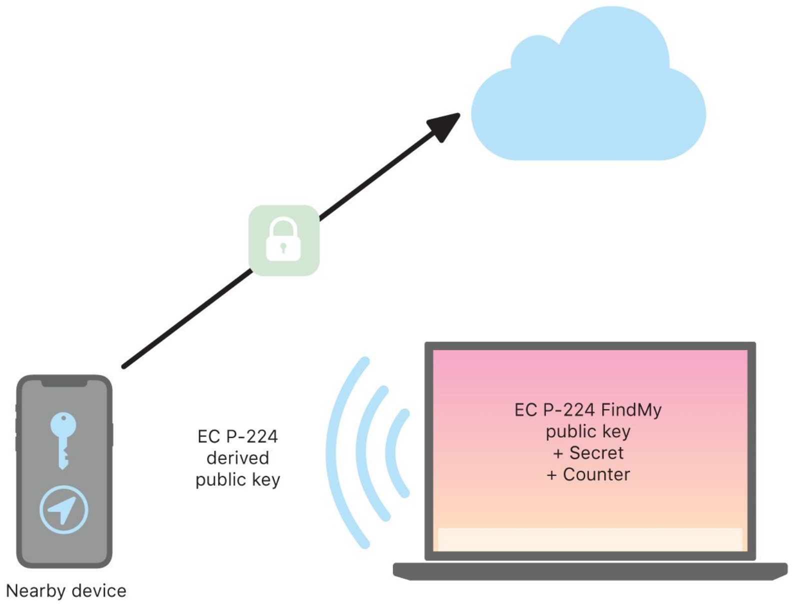
(can't fit all of the malicious use incidents here, this is just a sampling)

- 7/29/2019 Handoff All Your Privacy
- 1/6/2020 Discontinued Privacy
- 1/31/2020 furiousMAC/continuity Wireshark dissector presented at ShmooCon
- 4/26/2021 Who Can *Find My* Devices?
- 4/30/2021 AirTags released
- 5/1/2021 iFixit Teardown
- 5/13/2021 Adam Catley Reverse Eng Blog
- 7/16/2021 AirGuard Released on Google Play
- 9/12/2021 Woman reports AirTag under license plate cover
- 11/15/2021 Who Tracks the Trackers?
- 11/21/2021 Woman reports AirTag slipped into her bag
- 12/11/2021 Tracker Detect Released on Google Play
- 12/18/2021 Woman reports AirTag in wheel well, possible connection to luxury car theft
- 1/6/2022 Sports Illustrated model reports AirTag stalking
- 2/2/2022 AirTags spotted for sale online modified for no speaker
- 2/10/2022 "An update on AirTag and unwanted tracking"
- 2/16/2022 NY & PA AGs release consumer alerts about AirTag stalking

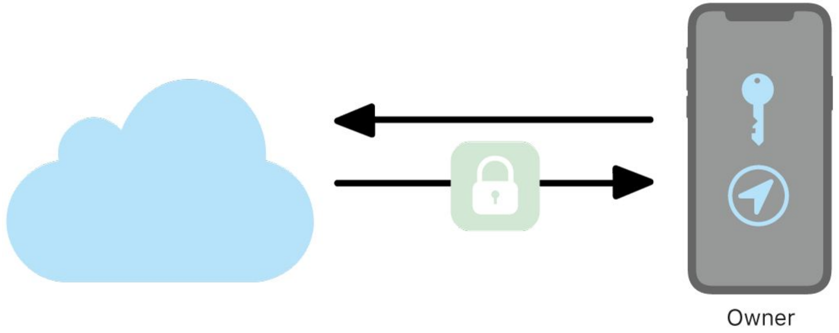
"Apple AirTags reportedly being used to stalk women — what to do" in *Tom's Guide*, by Paul Wagenseil, 12/20/2021.

"You can now buy 'silent' AirTags that won't beep — why that's dangerous" in Tom's Guide, by Paul Wagenseil, 2/3/2022.

System Architecture



How *Find My* locates devices

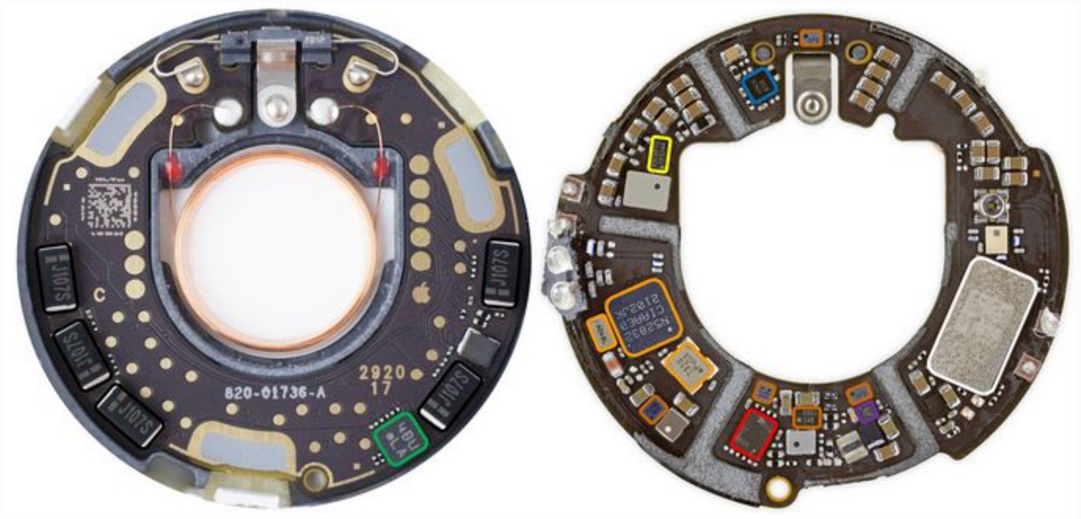


How the owner gets the device location from the Find My app

Source: Apple, "Using *Find My* to locate missing Apple devices". 2/18/2021

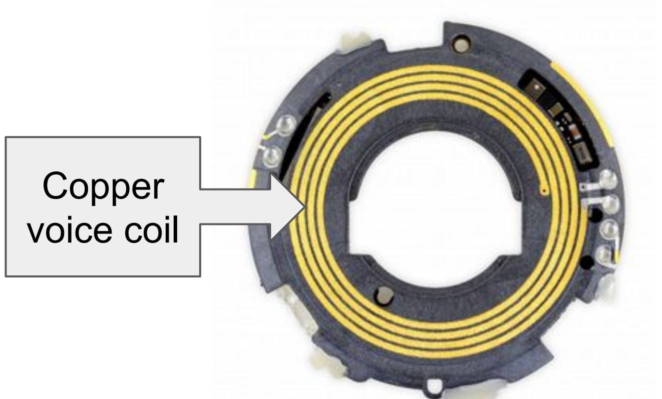
Hardware

PCB Overview



Source: Adam Catley, “Apple AirTag Reverse Engineering”, 5/13/2021

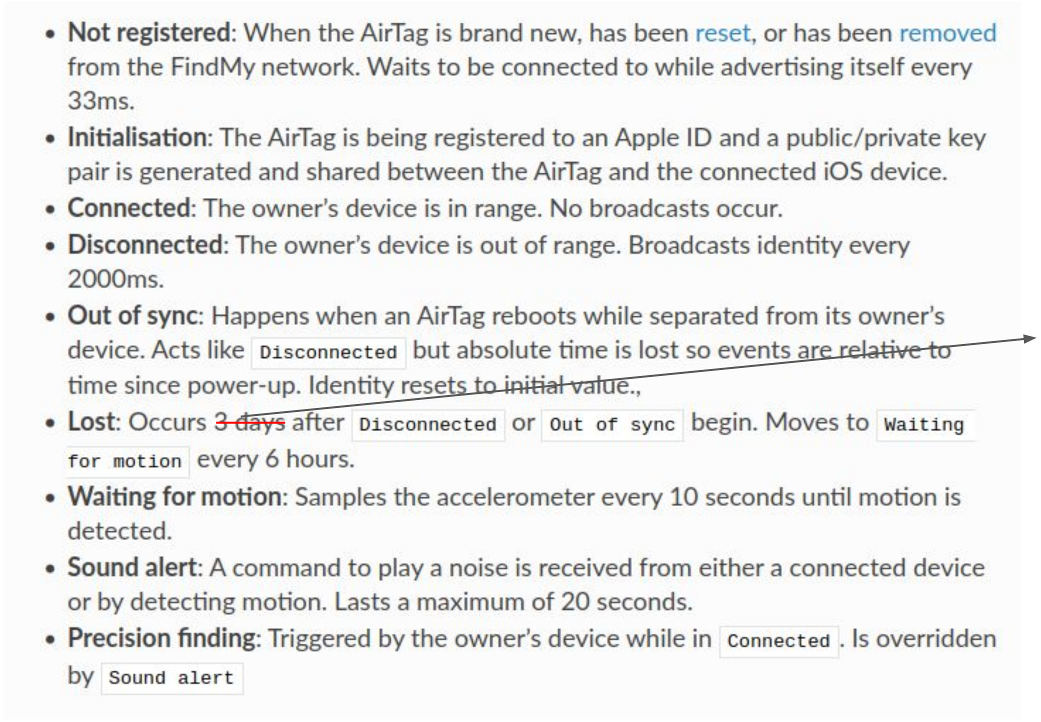
```
🟠 Nordic nRF52832 SoC with BLE and NFC, plus 32MHz and 32.768kHz crystals
⚪ Apple U1 UWB Transceiver
🔴 GigaDevice GD25LE32D 32Mbit NOR flash
🟢 Bosch BMA280 accelerometer
🔵 Maxim MAX98357AEWL audio amplifier
🟡 TI TPS62746 DC-DC buck converter
🟣 TI TLV9001IDPWR opamp
⚫ 100uF Electrolytic Capacitors (5x)
🟤 Unknown. Unable to decode markings
```



Source: iFixit, “AirTag Teardown: Yeah, This Tracks”, 5/1/2021

1. Bluetooth Low Energy (left) - 2.4GHz
2. NFC (middle) - 13.56MHz
3. Ultra-Wideband (right) - 6.5-8GHz

Device States



- **Not registered**: When the AirTag is brand new, has been reset, or has been removed from the FindMy network. Waits to be connected to while advertising itself every 33ms.
- **Initialisation**: The AirTag is being registered to an Apple ID and a public/private key pair is generated and shared between the AirTag and the connected iOS device.
- **Connected**: The owner's device is in range. No broadcasts occur.
- **Disconnected**: The owner's device is out of range. Broadcasts identity every 2000ms.
- **Out of sync**: Happens when an AirTag reboots while separated from its owner's device. Acts like `Disconnected` but absolute time is lost so events are relative to time since power-up. Identity resets to initial value.,
- **Lost**: Occurs ~~3 days~~ after `Disconnected` or `Out of sync` begin. Moves to `Waiting for motion` every 6 hours.
- **Waiting for motion**: Samples the accelerometer every 10 seconds until motion is detected.
- **Sound alert**: A command to play a noise is received from either a connected device or by detecting motion. Lasts a maximum of 20 seconds.
- **Precision finding**: Triggered by the owner's device while in `Connected`. Is overridden by `Sound alert`

random n in range: 8-24 hours

Source: Adam Catley, "Apple AirTag Reverse Engineering", 5/13/2021

BLE Message Structure: *Find My*

Field Name	Info	Example	Length	Type	Notes
btcommon.apple.findmy.status	Owner Connection & Battery Status	00	1	UINT8	Have only seen 0x00
btcommon.apple.findmy.publickey	Bytes 6-27 of Public Key	57364dc7fb77866c40c91076603cb37c1f59f923ab3a	22	Bytes	
btcommon.apple.findmy.publickey.bits	Bits 6-7 of Byte 0 of Public Key	03	1	UINT8	Only bits 0-1 used; Bits 2-7 are reserved
btcommon.apple.findmy.hint	Byte 5 of BT_ADDR of Primary key	00	1	UINT8	Have only seen 0x00
btcommon.apple.findmy.publickey.xcord	28-byte x-coord of Public Key	b953267519a8ef5b0bdea8bc5bf80bd0ee47e7d68b2bb8319cbbee0	28	STRING	

Contains code to add Apple Continuity protocol dissector to Wireshark. Caveat: compiling Wireshark from source code sucks and we haven't been able to do it yet

Source: https://github.com/furiousMAC/continuity

Advertisements

- AirTags broadcast our their public key in their BLE advertising packets
  - These are not authenticated
- Since location is determined based on the actual iPhones/Apple devices in the vicinity, if you can replay the advertisement packets in another location (and disable the original AirTag), then you can spoof the location

Y'all hear something?
I'm definitely an AirTag and I'm over here!
I 100% believe you and will do nothing to verify this information!
HackRF One
GREAT SCOTT GADGETS

Advertisements

- Typically, Bluetooth devices change the address they advertise from on regular intervals
    - This prevents devices (and therefore individuals) from being tracked using a single address
- For AirTags, the overall address and public key changes only once a day, but the last byte of advertisement data changes every 15 minutes
    - This means that you could still track an AirTag for at least a day just looking at the first portion of data
    - Additionally, this means if we want to spoof a packet, we have a 15 minute time window to do so

Prior Work

- These characteristics have been taken advantage of before to use the Apple FindMy network with fake AirTags
  - **OpenHaystack** - TU Darmstadt team reverse engineered the FindMy protocol and created a tool that allows users to create their own devices that leverage the FindMy network
  - **Send My** - using OpenHaystack to exfiltrate data from non-Apple devices through FindMy to the Apple cloud where it can be retrieved
  - **Find You** - using OpenHaystack to create a DIY AirTag that circumvents many of the protections “guaranteed” by Apple

OpenHaystack - Seemoo Lab, TU Darmstadt, https://github.com/seemoo-lab/openhaystack
Send My - Positive Security, https://positive.security/blog/send-my
Find You - Positive Security, https://positive.security/blog/find-you

Goals

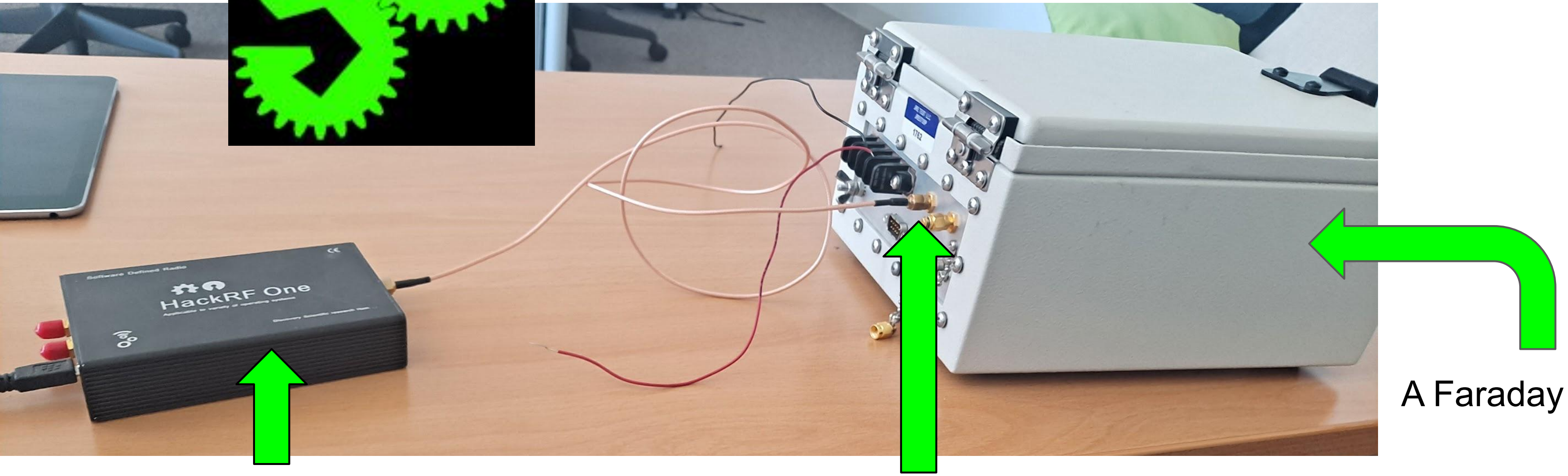
Breaking AirTag stuff so we can learn how to fix it or do it better

- Spoofing geolocation of AirTag →how do we report location from a low-power device in an untrustworthy environment?
- Replay/relay attack → how do we make this harder to do? Easier to detect or prevent?
- How do we make it easier for victims of tracker stalking to find devices fast and keep their locations from malicious individuals? Can we build on the work of AirGuard?

Bluetooth Programming on Linux with Your Computer’s Integrated Bluetooth Transceiver

- TLDR: don’t do it, if you can avoid it. Use a dongle or an SDR.
- But if you are gonna do it anyway:
  - The library you need to use is called BlueZ
  - Good tutorials:
    - https://www.bluetooth.com/bluetooth-resources/bluetooth-for-linux/
      - Pros: programming in Python
      - Cons: Requires name, email, and EULA to access materials, but they are free once you provide those.
    - https://people.csail.mit.edu/albert/bluez-intro/c404.html
      - Pros: Open access, no registration info required
      - Cons: programming in C

Using Software-Defined Radios



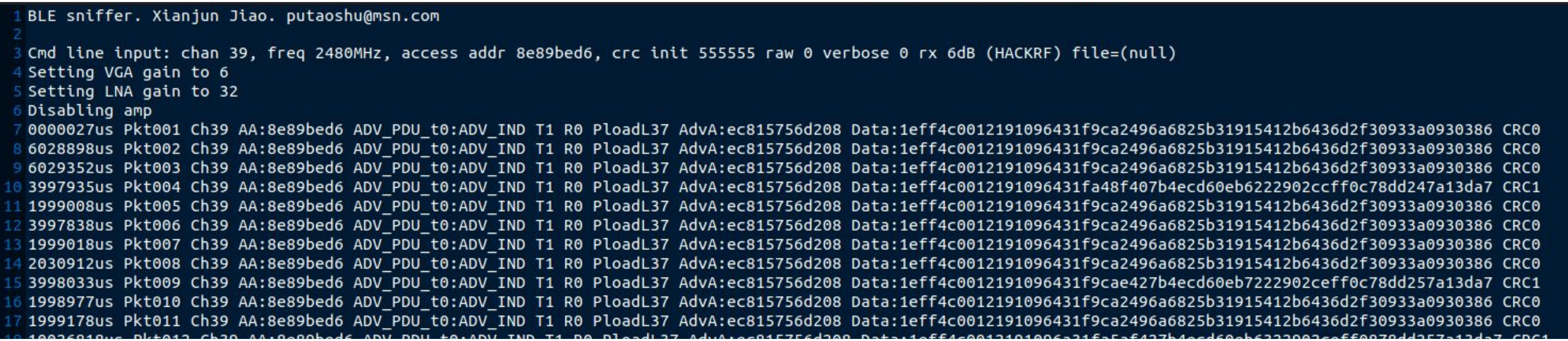
HackRF One SDR

Connection to antenna inside cage

A Faraday cage!

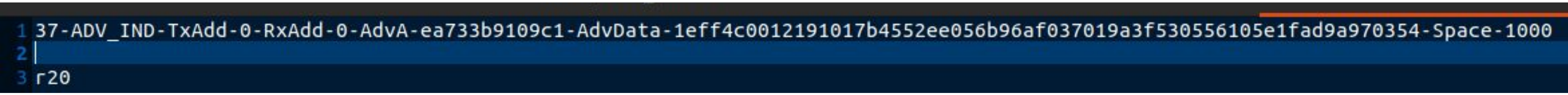
JiaoXianjun/BTLE

Receiver Mode



```
BLE sniffer. Xianjun Jiao. putaoshu@msn.com

Cmd line input: chan 39, freq 2480MHz, access addr 8e89bed6, crc init 555555 raw 0 verbose 0 rx 6dB (HACKRF) file=(null)
Setting VGA gain to 6
Setting LNA gain to 32
Disabling amp
0000027us Pkt001 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
6028898us Pkt002 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
6029352us Pkt003 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
3997935us Pkt004 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431fa48f407b4ecd60eb6222902ccff0c78dd247a13da7 CRC1
1999008us Pkt005 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
3997838us Pkt006 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
1999018us Pkt007 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
2030912us Pkt008 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
3998033us Pkt009 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9cae427b4ecd60eb7222902ceff0c78dd257a13da7 CRC1
1998977us Pkt010 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
1999178us Pkt011 Ch39 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL37 AdvA:ec815756d208 Data:1eff4c0012191096431f9ca2496a6825b31915412b6436d2f30933a0930386 CRC0
```

Transmission Mode



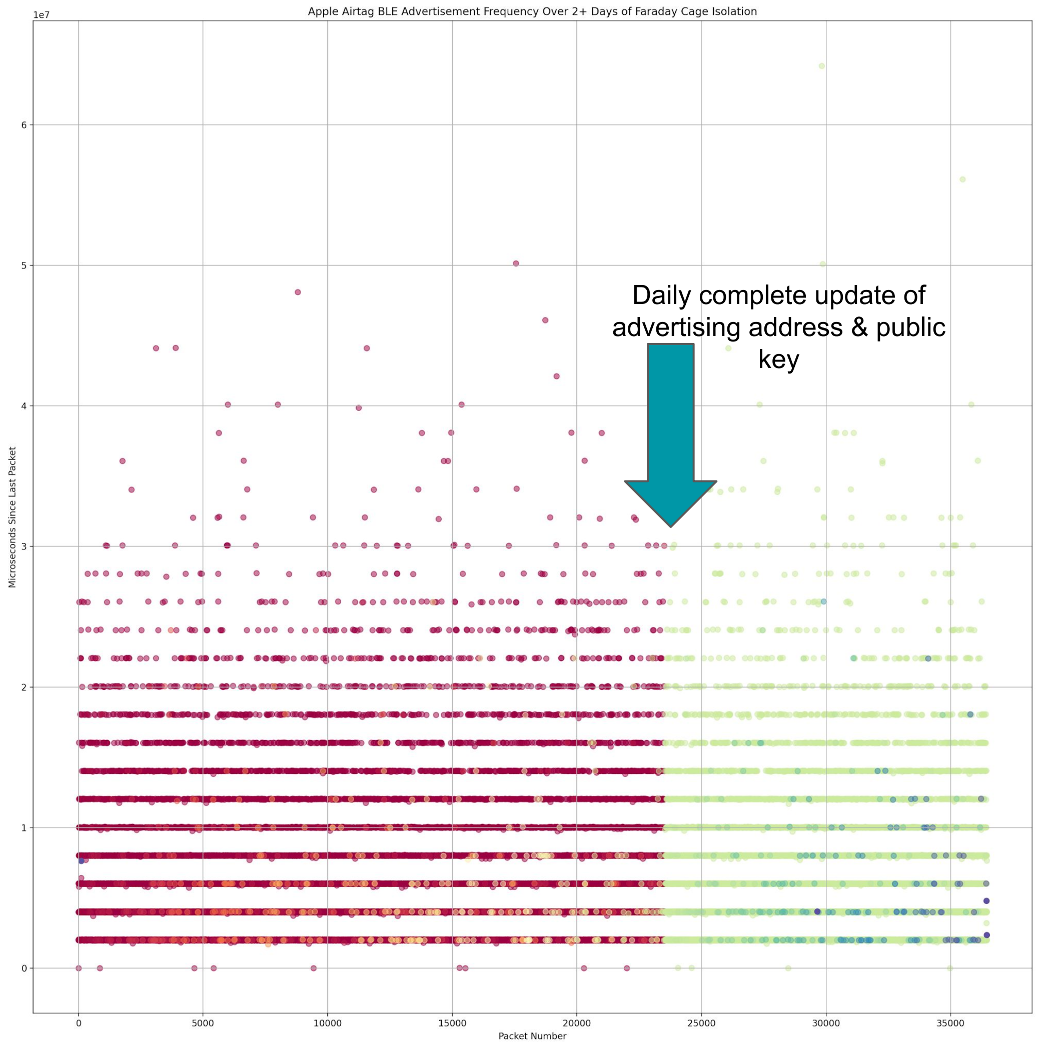
```
37-ADV_IND-TxAdd-0-RxAdd-0-AdvA-ea733b9109c1-AdvData-1eff4c0012191017b4552ee056b96af037019a3f530556105e1fad9a970354-Space-1000

r20
```

Apple Airtag BLE Advertisement Frequency Over 60+ Hours of Faraday Cage Isolation

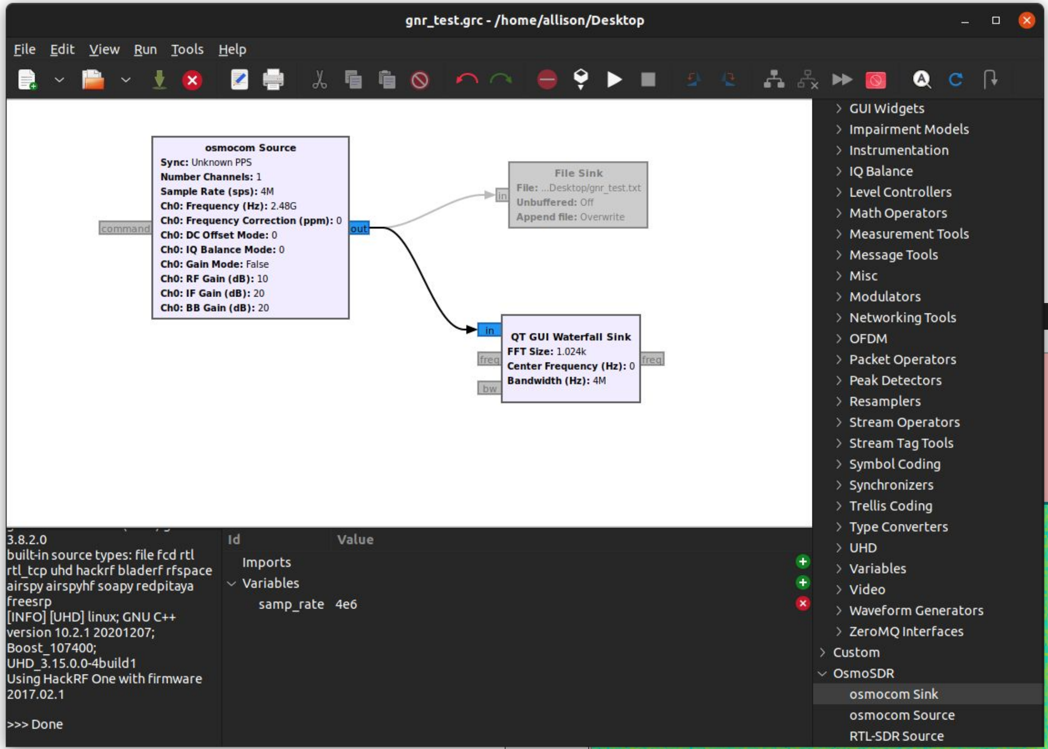
🔴🔵⚫⚪⇨data points colored by advertising address

X axis → packet number (in chronological order)

Y axis → microseconds since last packet received



GNU Radio



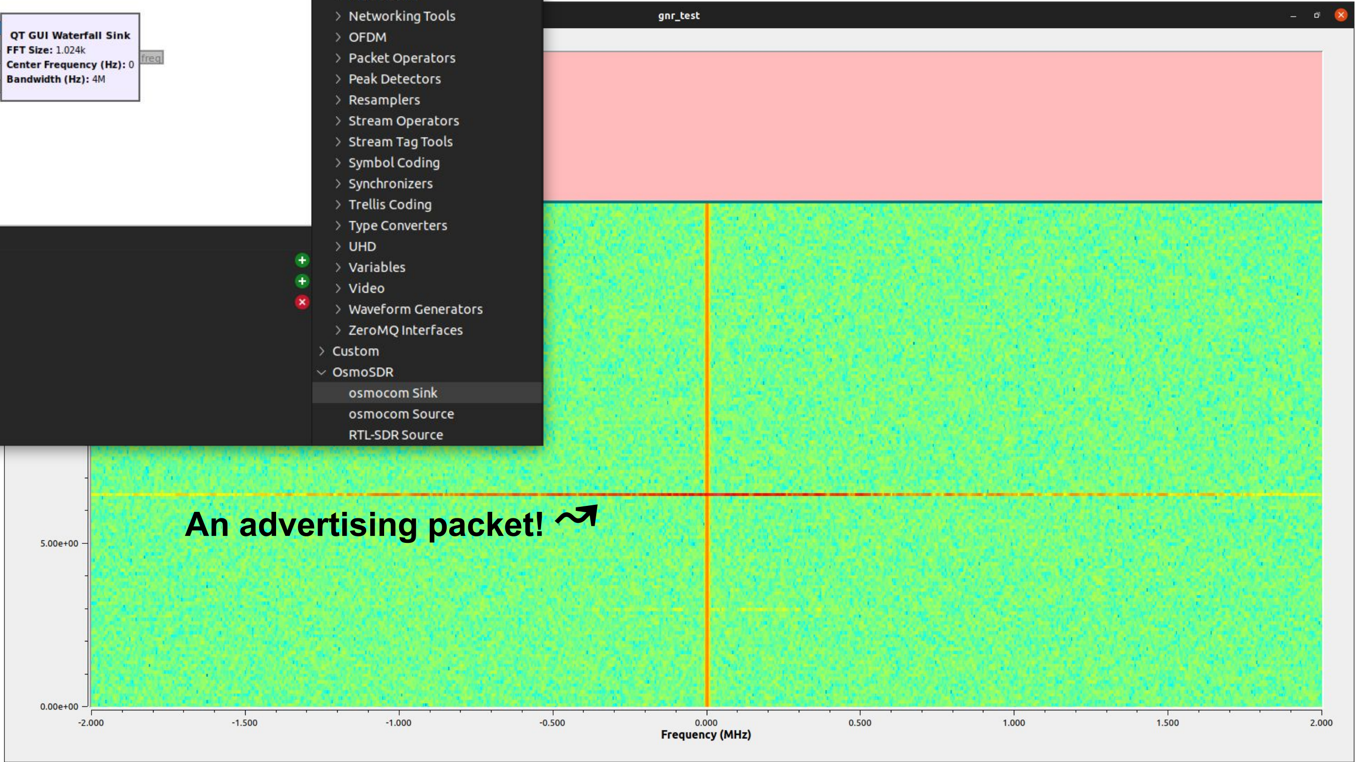
osmocom Source
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 4M
Ch0: Frequency (Hz): 2.48G
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

File Sink
File: ...Desktop/gnr_test.txt
Unbuffered: Off
Append file: Overwrite

QT GUI Waterfall Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 4M

```
3.8.2.0
built-in source types: file fcd rtl
rtl_tcp uhd hackrf bladerf rfspace
airspy airspyhf soapy redpitaya
freesrp
[INFO] [UHD] linux; GNU C++
version 10.2.1 20201207;
Boost_107400;
UHD_3.15.0.0-4build1
Using HackRF One with firmware
2017.02.1

>>> Done
```

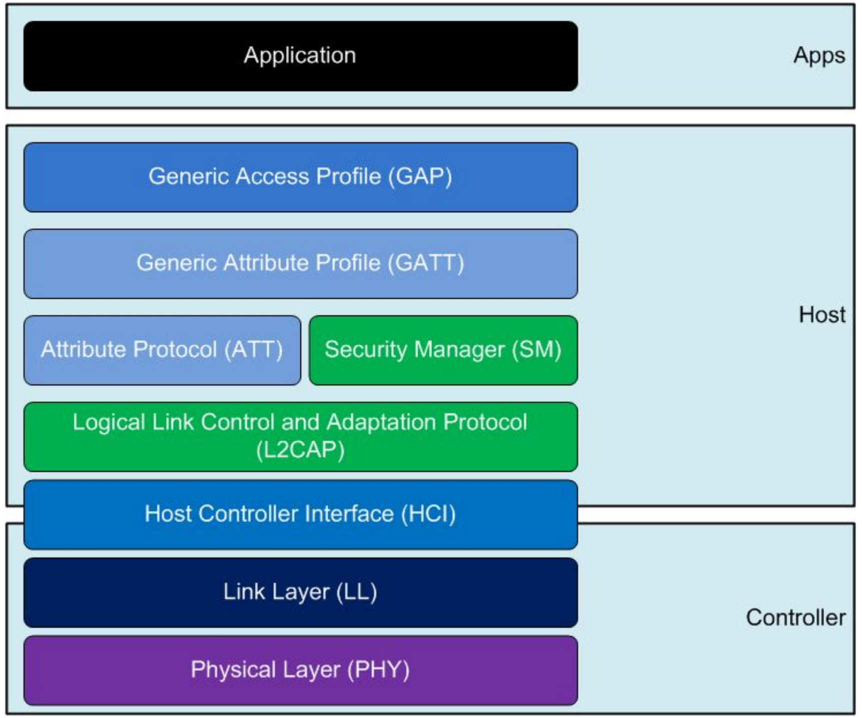
Id	Value
Imports	
Variables	
samp_rate	4e6



An advertising packet!

Can’t Get Smartphone Bluetooth Interfaces to Recognize Replayed Advertisements as Connectable Devices…Why?

- Current theory: advertising as a connectable device → other devices assume you should implement the Generic Attribute Profile protocol. When the SDR does not reply to Scan Requests as GAP specifies, consumer devices discard this device as a candidate
- Supporting evidence from experiment with earbud


Application
Apps
Generic Access Profile (GAP)
Generic Attribute Profile (GATT)
Attribute Protocol (ATT)
Security Manager (SM)
Host
Logical Link Control and Adaptation Protocol (L2CAP)
Host Controller Interface (HCI)
Link Layer (LL)
Physical Layer (PHY)
Controller
Source: Texas Instruments, SimpleLink Academy documentation

What’s Next

- Will our replay/relay attempt succeed if we mimic some Generic Access Profile behavior?
- Listening in on AirTag Ultra-Wideband spectrum activity
- Getting a spoofed GPS location accepted to *Find My*

References

Travis Mayberry, Ellis Fenske, Dane Brown, Jeremy Martin, Christine Fossaceca, Erik C. Rye, Sam Teplov, and Lucas Foppe. 2021. **Who Tracks the Trackers? Circumventing Apple's Anti-Tracking Alerts in the Find My Network**. In Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society (WPES '21). Association for Computing Machinery, New York, NY, USA, 181–186. DOI:https://doi.org/10.1145/3463676.3485616

Heinrich, Alexander, Stute, Milan, Kornhuber, Tim and Hollick, Matthias. **Who Can Find My Devices? Security and Privacy of Apple’s Crowd-Sourced Bluetooth Location Tracking System**. In Proceedings on Privacy Enhancing Technologies, vol.2021, no.3, 2021, pp.227-245. https://doi.org/10.2478/popets-2021-0045

Celosia, Guillaume and Cunche, Mathieu. **Discontinued Privacy: Personal Data Leaks in Apple Bluetooth-Low-Energy Continuity Protocols**. In Proceedings on Privacy Enhancing Technologies, vol.2020, no.1, 2020, pp.26-46. https://doi.org/10.2478/popets-2020-0003

👀👀 Forthcoming from Alexander Heinrich, Niklas Bittner, and Matthias Hollick: **AirGuard -- Protecting Android Users From Stalking Attacks By Apple Find My Devices**. Posted to arXiV on 2/23/2022, https://arxiv.org/abs/2202.11813

Furious MAC research group (https://furiousmac.com). https://github.com/furiousMAC/continuity . Initial commit January 29 2020, latest commit March 20 2021.

https://sites.google.com/ucsd.edu/airtag-blog/home

iFixit, “AirTag Teardown: Yeah, This Tracks”, 5/1/2021

Adam Catley, “Apple AirTag Reverse Engineering”, 5/13/2021

“Apple AirTags reportedly being used to stalk women — what to do” in Tom’s Guide, by Paul Wagenseil, 12/20/2021.

“You can now buy 'silent' AirTags that won't beep — why that's dangerous” in Tom’s Guide, by Paul Wagenseil, 2/3/2022.

Jiao Xianjun. https://github.com/JiaoXianjun/BTLE . Initial commit July 28 2014, latest commit July 16 2021.

OpenHaystack - Seemoo Lab, TU Darmstadt (https://github.com/seemoo-lab/openhaystack), Initial commit March 3 2021, latest commit January 4 2022

Send My - Positive Security (https://positive.security/blog/send-my), 5/12/21

Find You - Positive Security (https://positive.security/blog/find-you), 2/21/22

“All BLE guides are wrong (including this one)”. David Burnett, UC Berkeley, 3/6/2018.