Tracking You from a Thousand Miles Away! Turning a Bluetooth Device into an Apple AirTag Without Root Privileges

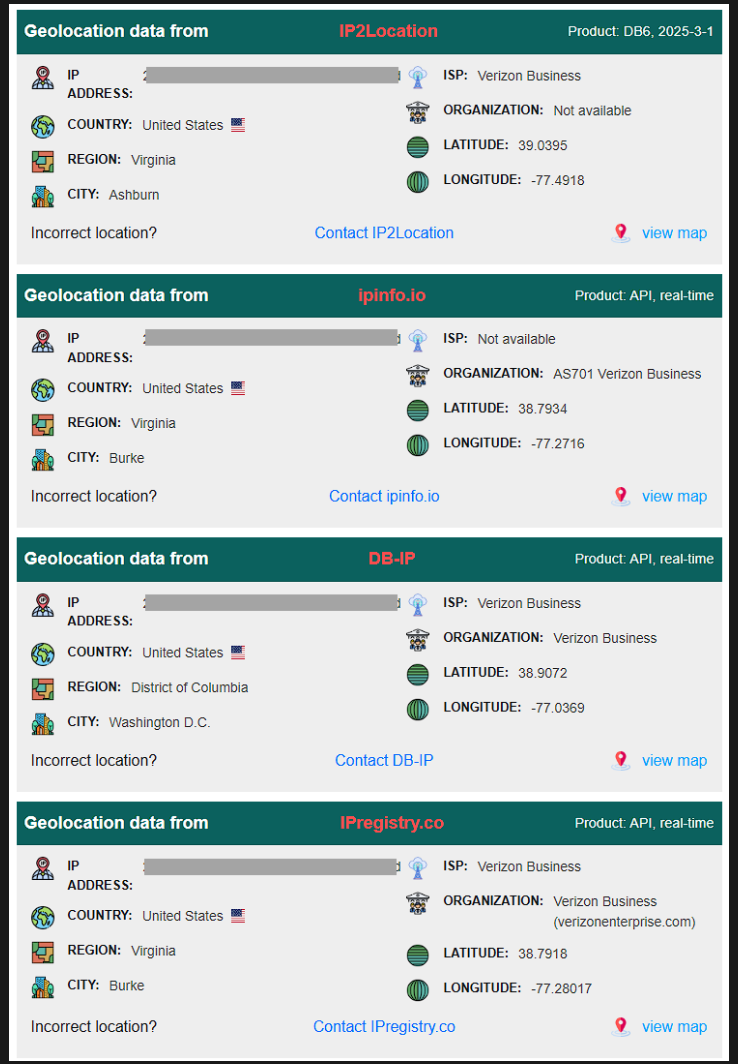
Junming Chen, Xiaoyue Ma, Lannan Luo, Qiang Zeng

34th USENIX Security Symposium
August 13–15, 2025
Seattle, WA, USA

Techniques for Tracking

- IP based Tracking

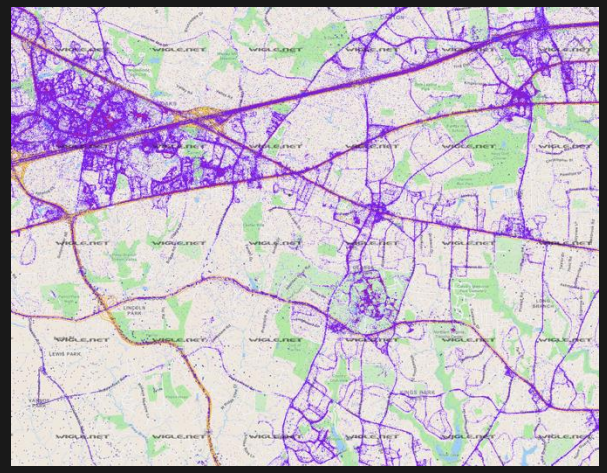
Accuracy	State
Latency	Hard to tell
Accessibility	ISP
Coverage	Determine by ISP

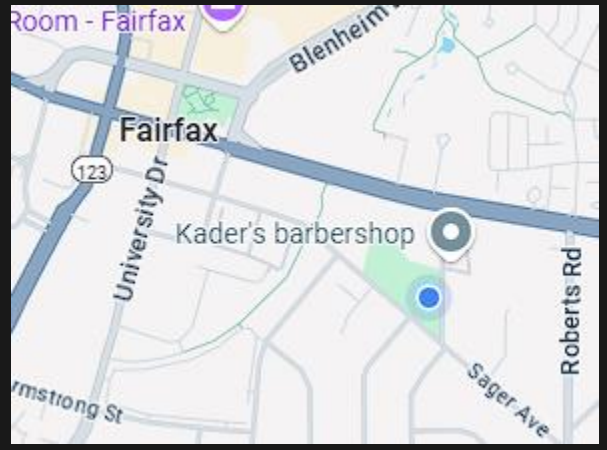


Techniques for Tracking

- Beacon / Wi-Fi based tracking

Accuracy	Street Blocks
Latency	Days, Months
Accessibility	Everyone (wigle.net)
Coverage	≤ Wi-Fi Coverage





Techniques for Tracking

- Crowdsourcing tracking

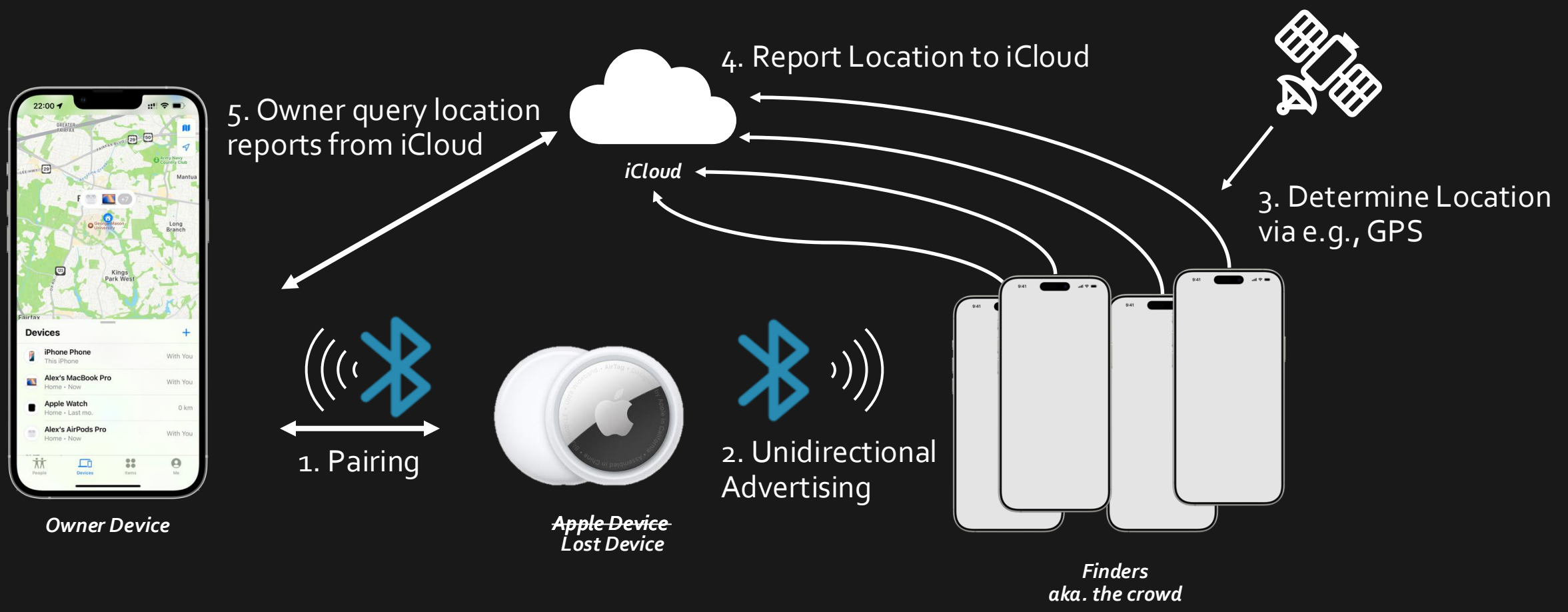
Protagonist of Today

Over **1.5 billion** iPhones forming a global tracking network

Who May Be Interested?

- ❑ Intelligence Departments
- ❑ Law Enforcement Agencies
- ❑ Advertising-Focused Mobile Apps and Software
- ❑ Cybercriminals

How Offline Finding Works?



The Explore Begins

Design: End-to-End Encryption, data exclusive to owner

Challenge: Put cryptographic key into limited payload space

Solution: Borrow extra space from BLE address

Addr. Type (2bits)

MAC address
6 bytes

11 2465B8942D55

Secp 224R1
Public Key

BLE Payload
31 bytes

... 1234567812345678 ...

The Explore Begins

BLE Address Types Overview *(from Internet)*

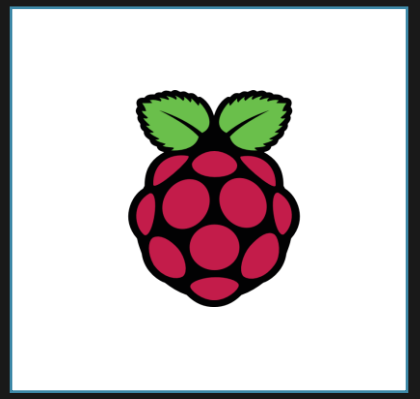
Types	MSB			LSB	
Public	OUI		Manufacturer Specific		Managed by IEEE
NRPA	0	0	Random part of NRPA		Randomly generated by device
RPA	0	1	Random part of prand	Hash	
Random Static	1	1	Random part of static address		Apple's choice

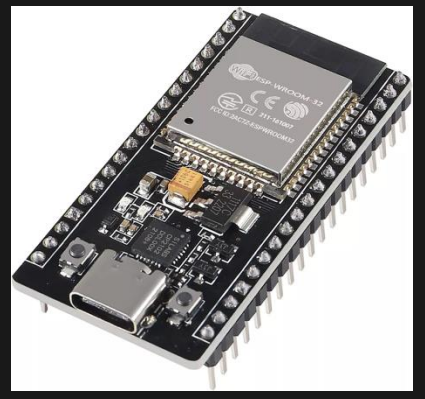
Only the public address can identify its assigned vendor!

The Explore Begins

OpenHayStack

- Pioneer research
- Based on reverse engineering
- Cheap and cool project for DIY





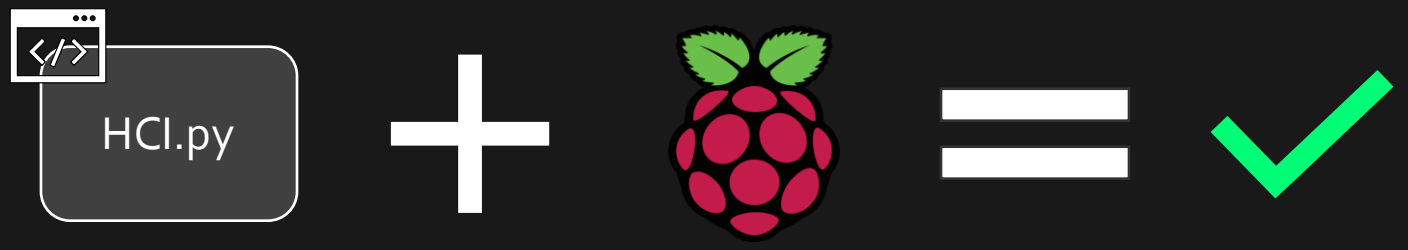
The Explore Begins

Replicating OpenHayStack

```
# ./hci.py --key 3q2+7xI0v52mn5n0uM7lgUooCh3JcbYMfN/JVQ==
key     (28) deadbeef1234 ...
```

Using Raspberry Pi

`Addr: DE:AD:BE:EF:12:34` ← Expected Result



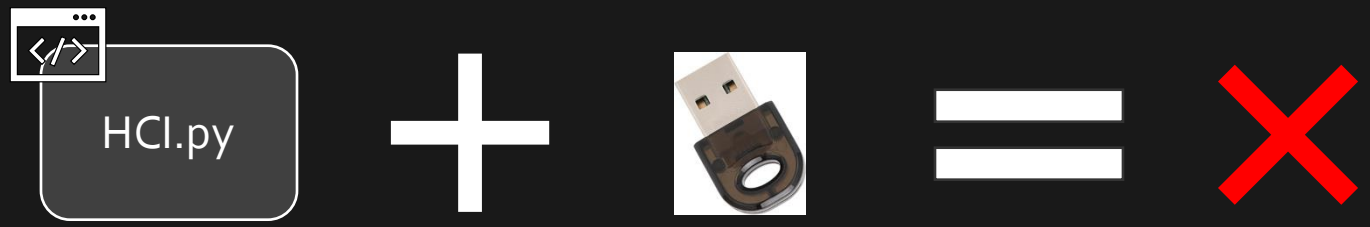
The Explore Begins

When checking address of adapter, we noticed something

```
# ./hci.py --key 3q2+7xI0v52mn5n0uM7lgUooCh3JcbYMfN/JVQ==
key     (28) deadbeef1234 ...
```

Using PC and Dongle

`Addr: 30:05:05:EC:A6:02` ← Unexpected Result



The Explore Begins

- Issues in OHS Code (*first*)

```
print(f"payload ({len(adv):2}) {adv.hex()}")

# Set BLE address
run_hci_cmd(["0x3f", "0x001"] + bytes_to_strarray(addr,
subprocess.run(["systemctl", "restart", "bluetooth"])
time.sleep(1)

# Set BLE advertisement payload
run_hci_cmd(["0x08", "0x0008"] + [format(len(adv), "x")]
```

Configure address

this is a cmd for changing Cypress adapter address

The Explore Begins

- Issues in OHS Code (*second*)

```
# Set BLE advertising mode
interval_enc = struct.pack("<h", interval_ms)
hci_set_adv_params = ["0x08", "0x0006"]
hci_set_adv_params += bytes_to_strarray(interval_enc)
hci_set_adv_params += bytes_to_strarray(interval_enc)
hci_set_adv_params += ["03", "00", "00", "00", "00", "00", "00", "00"]
hci_set_adv_params += ["07", "00"]
run_hci_cmd(hci_set_adv_params)
```

Own_Address_Type:

Value	Parameter Description
0x00	Public Device Address (**default**)
0x01	Random Device Address
0x02 – 0xFF	Reserved for future use

Bluetooth Specification

It conflicts with Apple's Spec!
Public != Static

The Explore Begins

- What's the ***address type*** for E4:65:B8:94:2D:55?

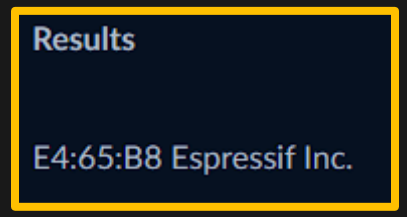
  Hint, E4 in binary is 11100100

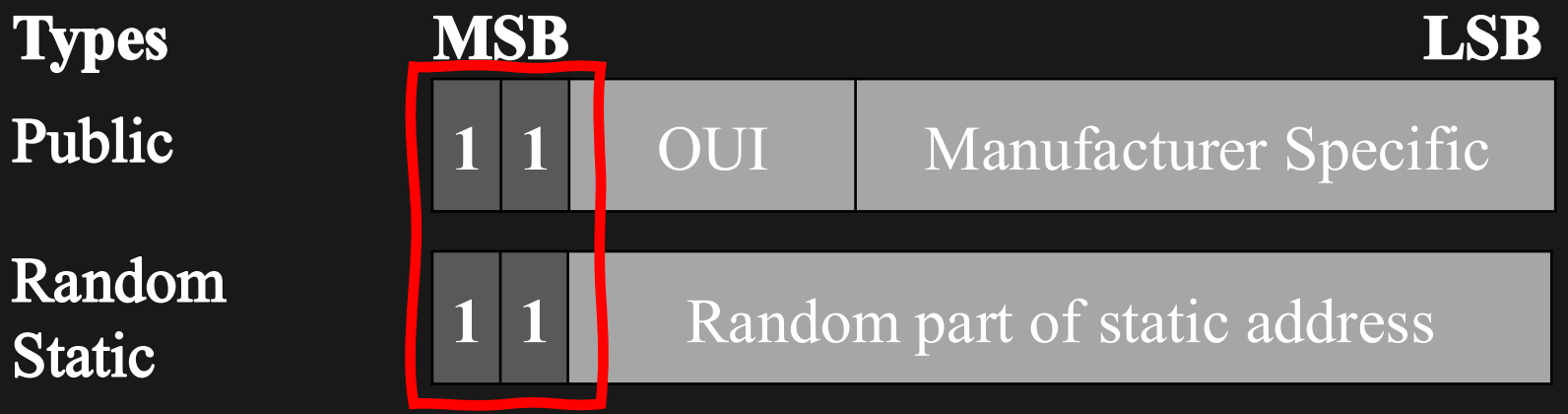
Types	MSB			LSB
Public	OUI		Manufacturer Specific	
NRPA	0	0	Random part of NRPA	
RPA	0	1	Random part of prand	Hash
Random Static	1	1	Random part of static address	

The Explore Begins

- Did you get it right?

Wireshark says





The Explore Begins

- BLE Address Types Overview *(the correct version)*

Types	TxAdd	MSB			LSB
Public	0	OUI			Manufacturer Specific
NRPA	1	0	0	Random part of NRPA	
RPA	1	0	1	Random part of prand	Hash
Random Static	1	1	1	Random part of static address	

TxAdd is required to determine address type!

The Explore Begins

Find My network accepts more address types!

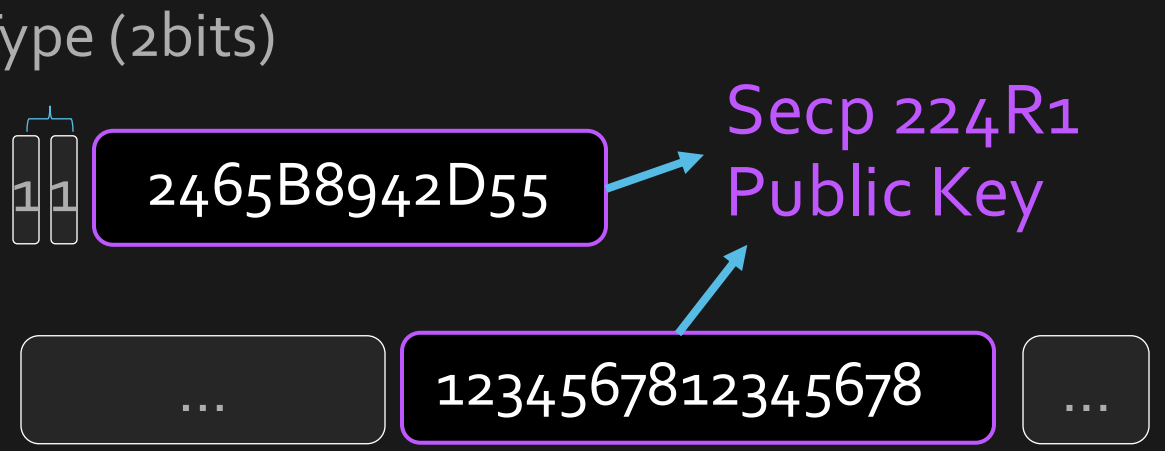
Apple, OpenHayStack, and derived works all missed this.

Operating System	Address Type
Linux	Public Address
Android	RPA
Windows	NRPA

Unprivileged Address Types

The Explore Begins

Splitting Public Key and Use as MAC Address



Exploit Deep Dive

Tailored Attacks

- Attack for ***Linux***
- Attack for ***Everything***

Exploit Deep Dive

➢ Attack for ***Linux***

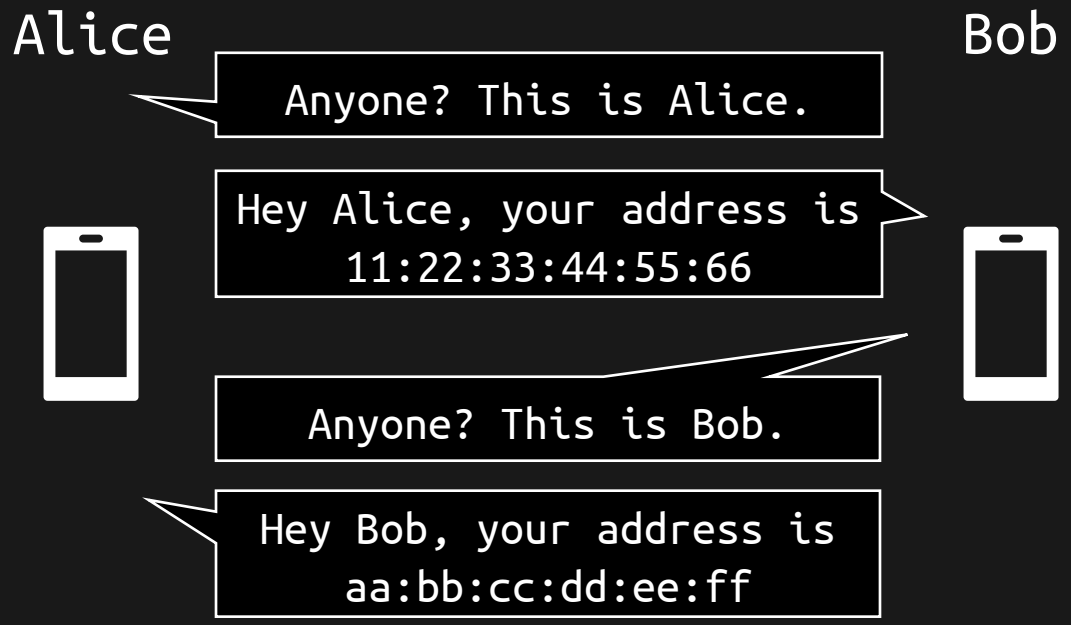
Linux uses BlueZ ➡ BlueZ Adv. with Public Addr. ➡ Only 47,054 OUI

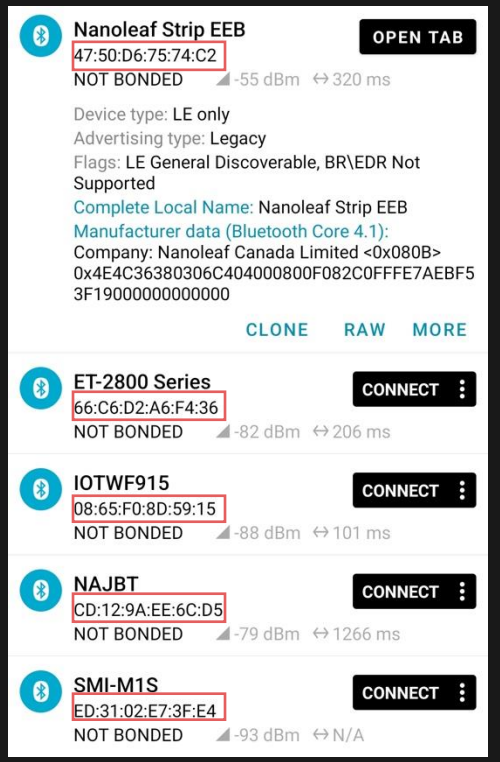
Rainbow Table!

Total space: 20.x TB

Exploit Deep Dive

- Attack for ***Everything***
  - Android prevented Bluetooth tracking
  - Allowed reading address from received Msg.





nRF connect shows address of surrounding devices

Benchmark

Attacks should be practical in dimensions of time and cost

Model	KPS In Billion	Purchase Price	Price-to-KPS Ratio*	Rent Price	Rent-to-KPS Ratio*
RTX 3070	2.64	$499	189	$0.13	0.050
RTX 3080	4.90	$699	**143**	$0.22	**0.045**
RTX 4090	8.22	$1,599	195	$0.40	0.049
A100 80GB	9.07	$13,224	1458	$1.60	0.178
H100 80GB	11.91	$48,200	4047	$2.14	0.180

*the smaller the more efficient (cheaper).

Benchmark

2.76 Estimated Minutes

90% possibility finding a match using 200 RTX 3080

2.20 Estimated USD

for renting RTX3080

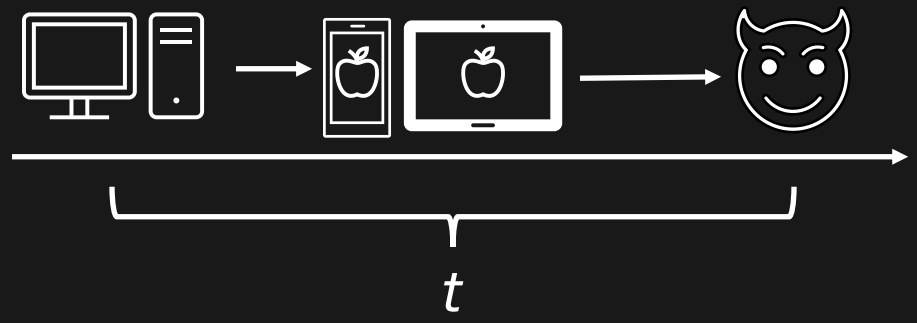
66 Measured Seconds

in reality using 200 units

For a few targets, Don’t Store, Don’t Buy, Just Rent!

Benchmark

Latency Measurement



Avg. 6.75 minutes

In University Campus

Avg. 8.09 minutes

At Single Family House

Performs just like a normal AirTag

Benchmark

Accuracy Measurement

Avg. **9.8** feet

At single family house

229.8 feet

During a flight from DC to LAX
at 536.243 mph

Closing Remarks

Security Adversary

Apple acknowledged our contribution and released patches in December 2024 to fix this vulnerability. Including,

iOS	18.2	tvOS	18.2
visionOS	2.2	macOS Ventura	13.7.2
iPadOS	17.7.3 & 18.2	macOS Sonoma	14.7.2
watchOS	11.2	macOS Sequoia	15.2

We speculate the vulnerability will take years to eventually fade out.