

Outsmarting Bluetooth Smart

Mike Ryan
iSEC Patners

CanSecWest
Mar 14, 2014

Quick Note

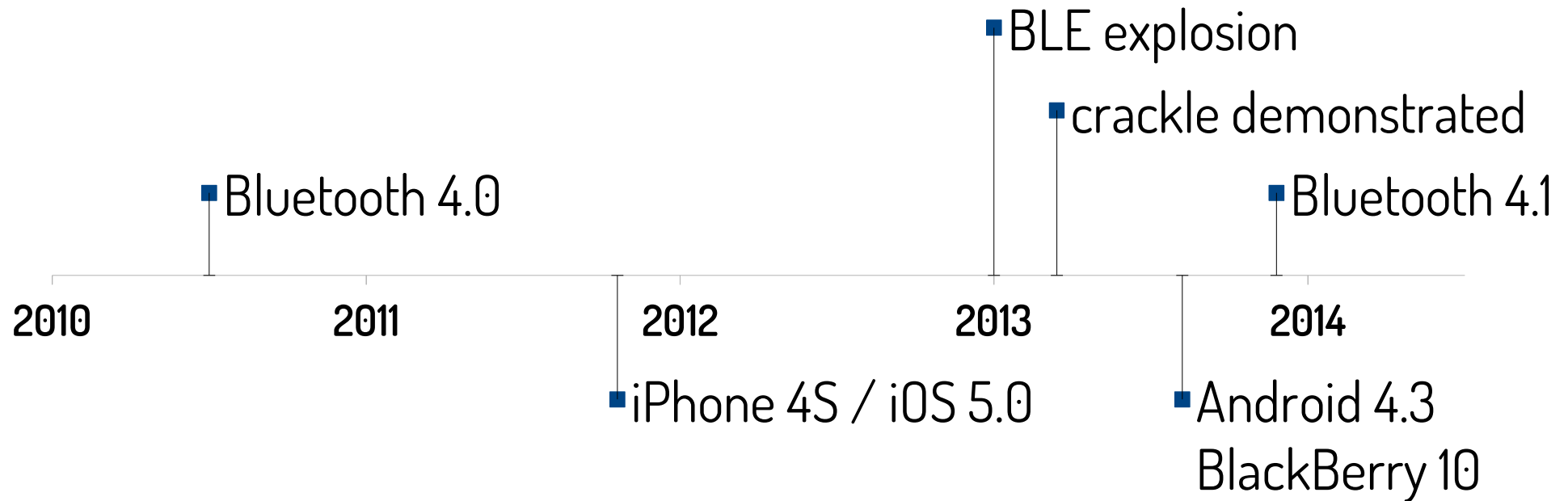
Bluetooth Smart

Bluetooth Low Energy

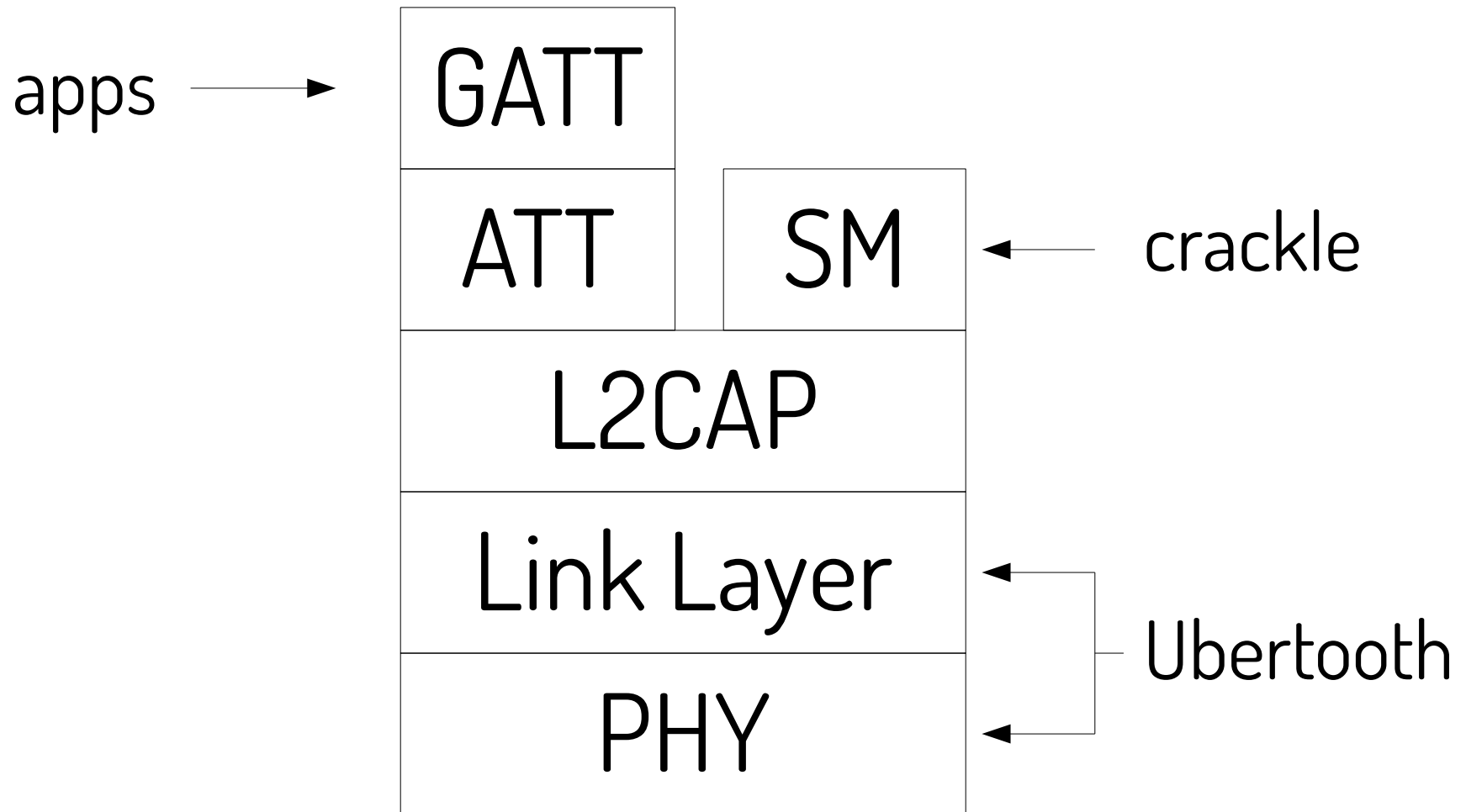
BLE

all the same thing!

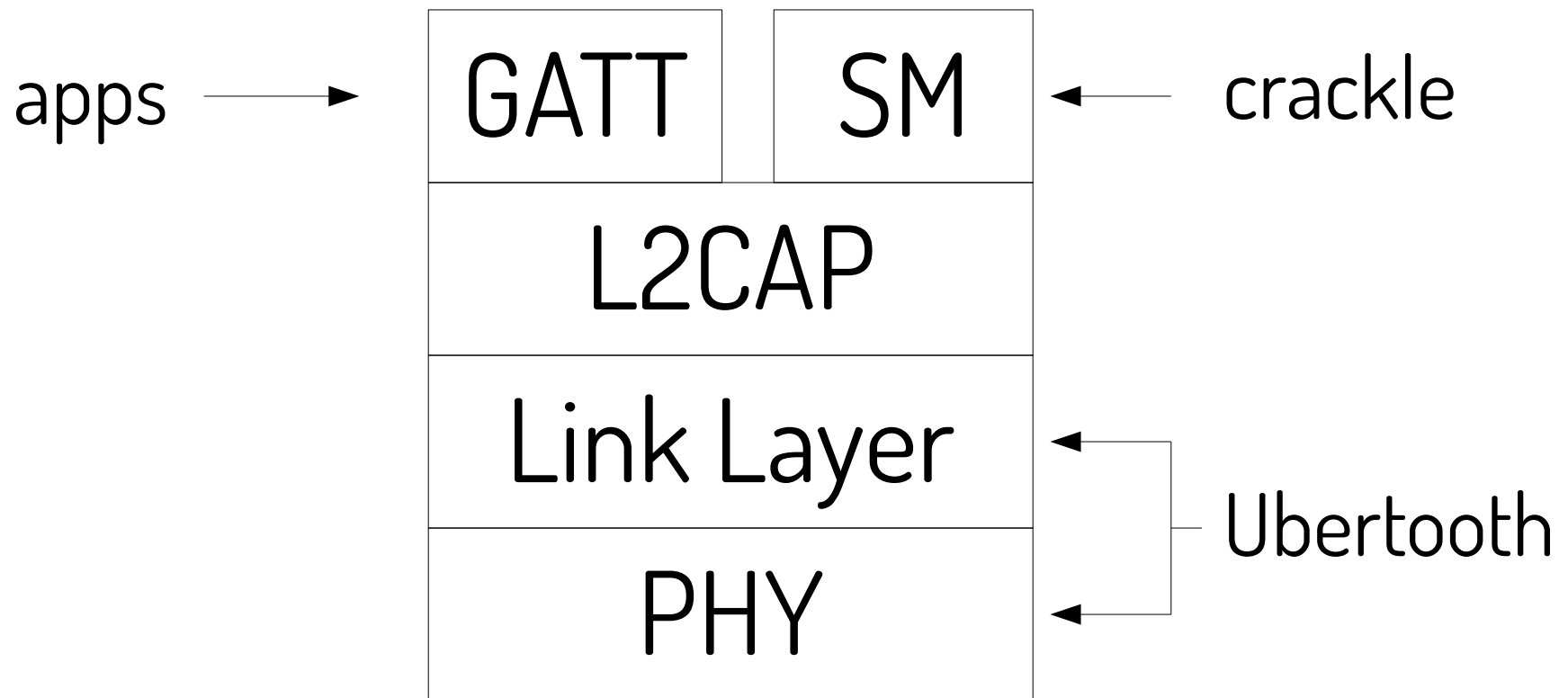
A Brief History of BLE



BLE Stack



Simplified BLE Stack



GATT: Characteristics

- Name – value pair
- With certain operations

Temperature – read

Lightbulb illumination – write

Implementation detail: name is UUID

GATT: Services

→ Group of related characteristics

Heart rate service

Temperature service

Device information service

Talking to Devices

- LightBlue
- gatttool

Peripheral: **Nike+ FuelB**

[Clone Peripheral](#)

Services Found

Device Information

UUID: 180A
0x83CDC410-31DD-11E2-81C1-0800
UUID: 83CDC410-31DD-11E2-81C1-080

Peripheral: **Nike+ Fue**
Service: **Device Inf**

Characteristics Found

Manufacturer Name St

UUID: 2A29 Properties: Read

Model Number String

UUID: 2A24 Properties: Read

Serial Number String

UUID: 2A25 Properties: Read

Firmware Revision Stri

UUID: 2A26 Properties: Read

Software Revision Stri

UUID: 2A28 Properties: Read

Hardware Revision Str

UUID: 2A27 Properties: Read

Peripheral: **Nike+ FuelBand SE**

Service: **Device Information**

Characteristic: **Manufacturer Name String**

Data Type: **string**

ASCII Nike
Hex 0x4E696B65
Decimal 1701538126
Date 2014/03/12 16:24:36:169

Read

Goal: Understand a Device

- 1) Sniff it
- 2) Connect with gatttool or LightBlue
- 3) Dump HCI using hcidump
- 4) Disassemble the app
- 5) Clone the device

Clues in the App

```
MOV      R0, #(selRef_UUIDWithString_ - 0xD2EB0) ; selRef_UUIDWithString_  
MOV      R8, #(classRef_CBUUID - 0xD2EB2) ; classRef_CBUUID  
ADD      R0, PC ; selRef_UUIDWithString_  
ADD      R8, PC ; classRef_CBUUID  
LDR      R5, [R0] ; "UUIDWithString:"  
LDR.W    R0, [R8] ; _OBJC_CLASS_$_CBUUID  
MOV      R2, #(cfstr_86130247E942_1 - 0xD2EC2) ; "8[REDACTED]2-4FE5-AA46-111111111111"  
ADD      R2, PC ; "8[REDACTED]2-4FE5-AA46-111111111111"  
MOV      R1, R5  
BLX      __CBUUID_UUIDWithString__  
MOV      R7, R7  
BLX      _objc_retainAutoreleasedReturnValue  
MOV      R1, #(_OBJC_IVAR_$_BluetoothManager.[REDACTED]ServiceUUID - 0xD2ED8) ; CBUUID [REDACTED]
```

“8XXXXXX-XXX2-4FE5-AA46-111111111111”

“0x0979”

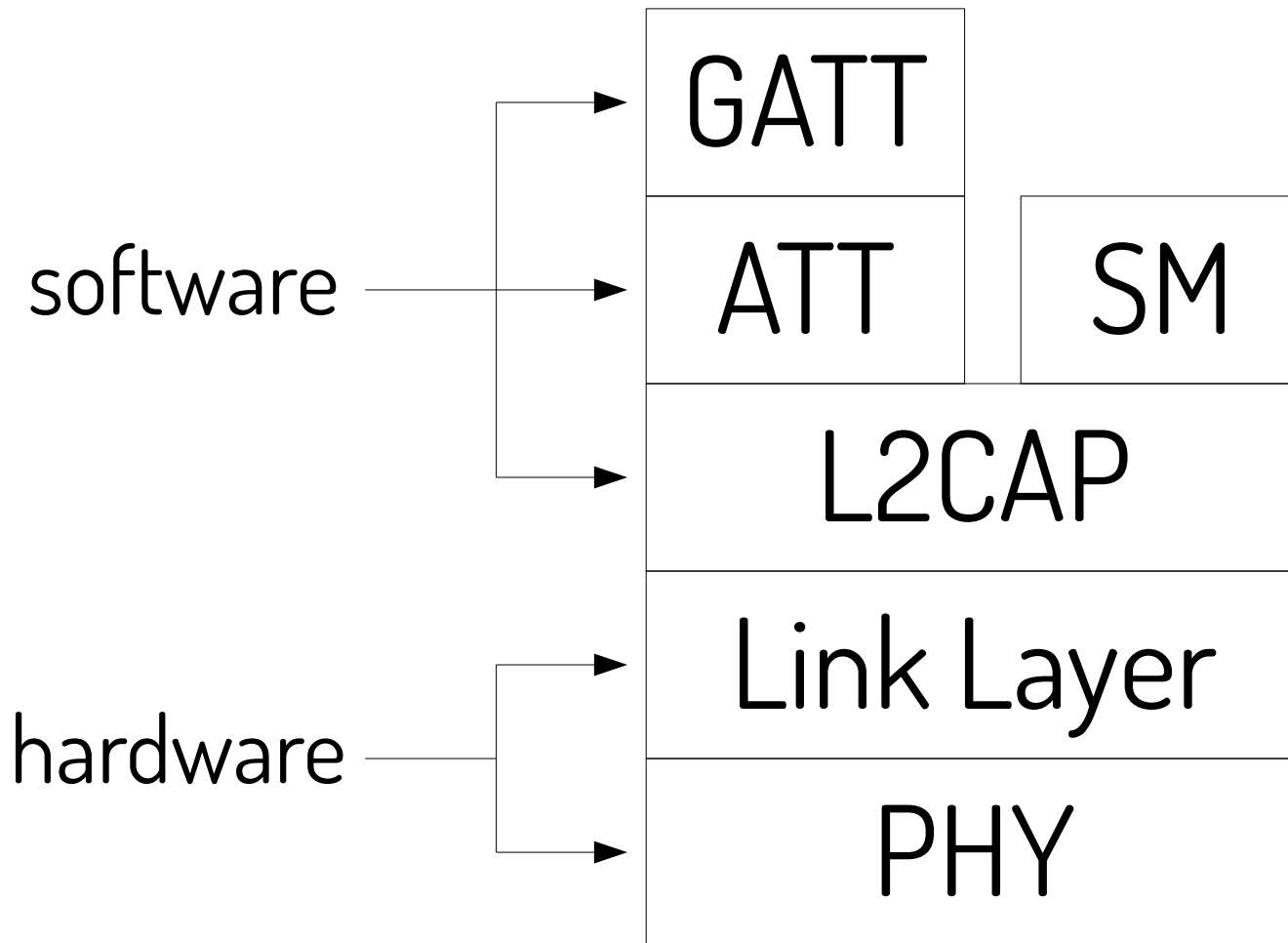
CBUUID *xxxxxxDataStreamCharUUID

```
MOV      R1, R5  
ADD      R2, PC ; "0x0979"  
BLX      __CBUUID_UUIDWithString__  
MOV      R7, R7  
BLX      _objc_retainAutoreleasedReturnValue  
MOV      R1, #(_OBJC_IVAR_$_BluetoothManager.[REDACTED]DataStreamCharUUID - 0xD2F0A) ; CBUUID  
ADD      R1, PC ; CBUUID * [REDACTED]DataStreamCharUUID;  
LDR      R6, [R1] ; CBUUID * [REDACTED]DataStreamCharUUID;  
LDR      R1, [R4, #61]
```

Clone the Device

- BLE devices are role-flexible
- This includes BlueZ

BLE Stack



Length Fields Aplenty

00 17 XX XX XX 22 00 02 01 06
03 02 0a 18 06 ff 6b 00 01 00 00
02 0a 00

advertising

06 09 05 00 04 00 cc 20 00 2c 00

data

HCI H4

The image shows a Wireshark network capture of Bluetooth HCI H4 traffic. The main pane displays a list of 14 packets (No. 21-34) with columns for No., Time, Source, Destination, Protocol, Length, and Info. Packet 27 is highlighted, showing a 'Sent Set Event Mask' command. The packet details pane for packet 27 shows the command opcode and parameters for the Set Event Mask command, including flags for Inquiry Complete, Inquiry Result, Connect Complete, and Connect Request.

No.	Time	Source	Destination	Protocol	Length	Info
21	0.000273			HCI_CMD	11	Sent Delete Stored Link Key
22	0.000695			HCI_EVT	9	Rcvd Command Complete (Delete Stored Link Key)
23	0.000219			HCI_CMD	4	Sent LE Read Buffer Size
24	0.000755			HCI_EVT	10	Rcvd Command Complete (LE Read Buffer Size)
25	0.000184			HCI_CMD	4	Sent LE Read Advertising Channel Tx Power
26	0.000816			HCI_EVT	8	Rcvd Command Complete (LE Read Advertising Channel)
27	0.000150			HCI_CMD	12	Sent Set Event Mask
28	0.000837			HCI_EVT	7	Rcvd Command Complete (Set Event Mask)
29	0.000248			HCI_CMD	12	Sent LE Set Event Mask
30	0.000768			HCI_EVT	7	Rcvd Command Complete (LE Set Event Mask)
31	0.000154			HCI_CMD	4	Sent Read Local Supported Commands
32	0.004862			HCI_EVT	71	Rcvd Command Complete (Read Local Supported Command)
33	0.000164			HCI_CMD	5	Sent Write Simple Pairing Mode
34	0.061811			HCI_EVT	7	Rcvd Command Complete (Write Simple Pairing Mode)

Frame 27: 12 bytes on wire (96 bits), 12 bytes captured on interface (96 bits) on interface 0
Bluetooth HCI H4
Bluetooth HCI Command - Set Event Mask
Command Opcode: Set Event Mask (0x0c01)
Parameter Total Length: 8
.... 1 = Inquiry Complete: true (0x01)
.... .1. = Inquiry Result: true (0x01)
.... .1.. = Connect Complete: true (0x01)
.... 1... = Connect Request: true (0x01)

```
0000 01 01 0c 08 ff ff fb ff 07 f8 bf 3d
```

Fuzz Platform: Linux

- Raw HCI (somehow?)
- BlueZ

Raw HCI: HCI_USER_SOCKET

- SOCK_RAW for Bluetooth
- In Linux 3.13

Scapy-based fuzzer

- Scapy rules
- No seriously, it rules

- Status
 - Able to establish connections
 - Basic communication
 - Generative fuzzing using Scapy's fuzz()

HCI_USER_SOCKET Availability

- Right now, at this moment:
 - Fedora 20
 - Arch (since Jan or Feb)
 - Debian Unstable
- Coming soon:
 - Ubuntu Trusty (next month)
 - Debian Testing (probably)
- Others:
 - Kali: nope, 1.0.6 is still 3.12
 - Pentoo: liveCD is older, but you can install ≥ 3.13 from Portage

Attack Platform: Linux BlueZ

- Very good code!
- Mutative fuzzing

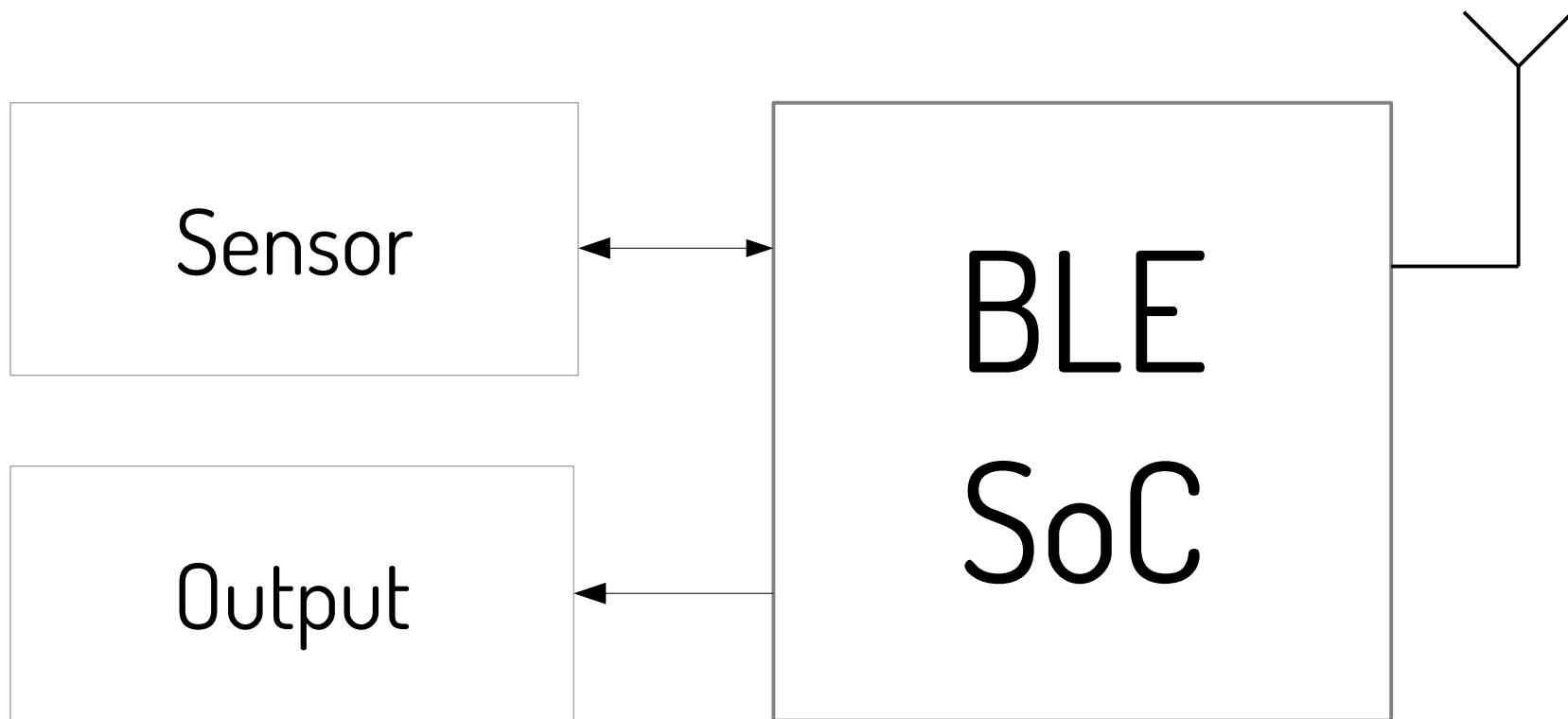
gatttool scripting
ftw!

```
guint g_attrib_send(...) {  
    ...  
  
    opcode = pdu[0];  
  
    c->opcode = opcode;  
    c->expected = opcode2expected(opcode);  
    c->pdu = g_malloc(len);  
  
    fuzz(pdu, len);  
  
    memcpy(c->pdu, pdu, len);  
}
```

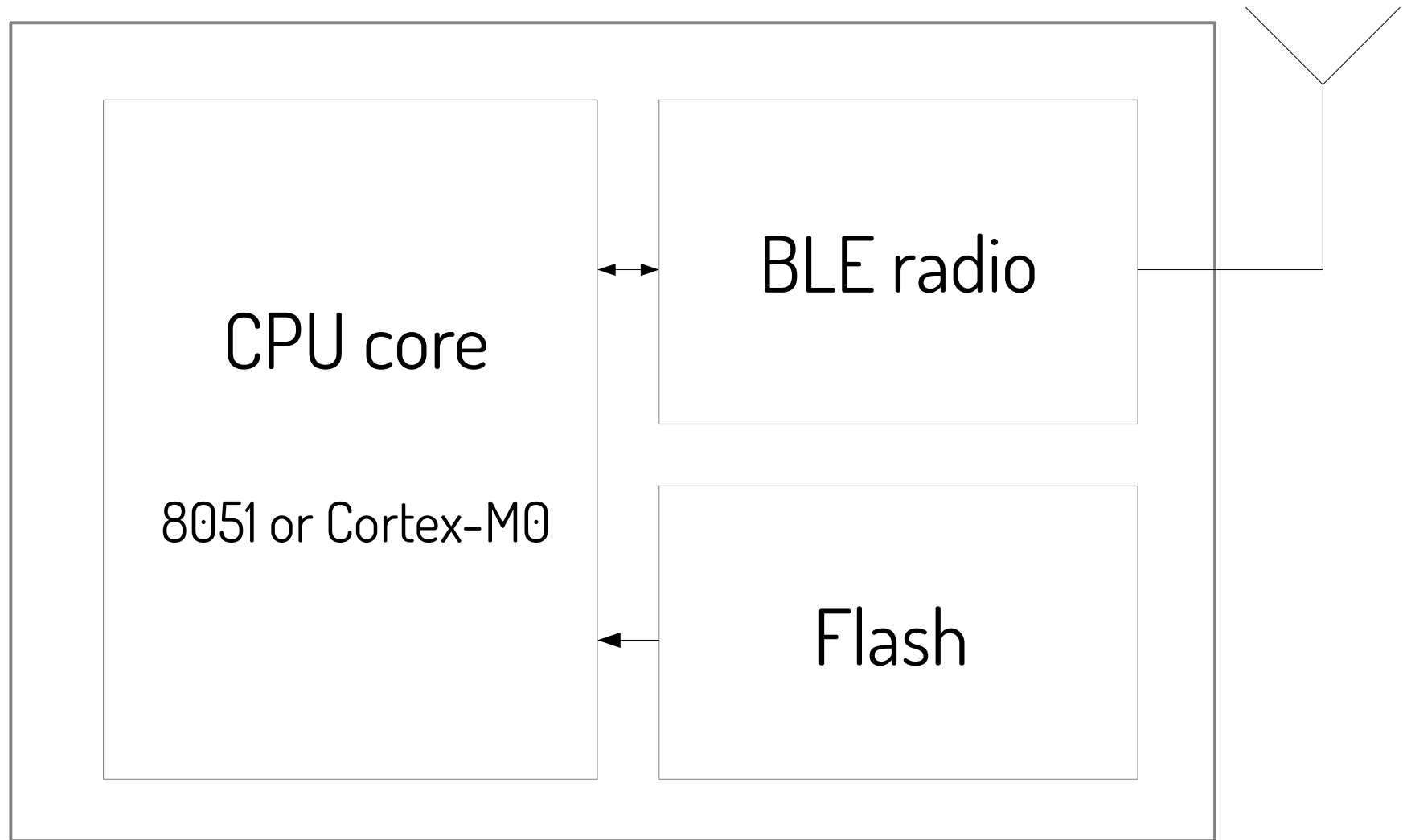
~~Victims~~ Targets

- Smartphones
 - Android: Bluedroid
 - iOS
 - Windows Phone 8.1
 - BlackBerry 10
- Devices
 - TI
 - Nordic

Generic BLE Device



BLE System-on-Chip



Primary Target: Bluedroid

```
~/bluedroid $ git grep memcpy | wc -l  
729
```


Bluedroid on the Phone

```
$ ps | grep bluetooth  
bluetooth 13201 871 862128 28940 ffffffff \\  
00000000 S com.android.bluetooth
```

CAP_NET_ADMIN: Almost as good as root

Source: Nexus 4 running Android 4.3

CAP_NET_ADMIN

- Perform various network-related operations:
 - interface configuration;
 - administration of IP firewall, masquerading, and accounting
 - modify routing tables;
 - bind to any address for transparent proxying;
 - set promiscuous mode;
 - use `setsockopt(2)` to set the following socket options:
SO_DEBUG...

Source: `capabilities(7)` man page

Fuzz process

- Script gatttool to send lots of data
- Use modified fuzzing BlueZ
- Log packets with btmon
- Watch adb logcat for crash
- Tweak BlueZ to send evil packet once it's identified

DEMO

→ DEMO

→ demo

- demo

→ demo

→ DeMo

→ Demoooooooooooooooooooo

Vuln Details

```
F/libc (19174): FORTIFY_SOURCE: memcpy buffer overflow. Calling abort().
F/libc (19174): Fatal signal 11 (SIGSEGV) at 0xdeadbaad (code=1), thread 19956
```

```
void gatt_process_notification(tGATT_TCB *p_tcb, UINT8 op_code,
                              UINT16 len, UINT8 *p_data)
{
    ...
    GATT_TRACE_DEBUG0("gatt_process_notification ");

    STREAM_TO_UINT16 (value.handle, p);
    value.len = len - 2;
    memcpy (value.value, p, value.len);
}
```

FORTIFY_SOURCE

```
extern "C" void *__memcpy_chk(void *dest, const void *src,
                              size_t copy_amount, size_t dest_len)
{
    if (__predict_false(copy_amount > dest_len)) {
        __fortify_chk_fail("memcpy buffer overflow",
                           BIONIC_EVENT_MEMCPY_BUFFER_OVERFLOW);
    }

    return memcpy(dest, src, copy_amount);
}
```

Timeline

- 2013-09-30: Notified Google
- 2013-10-07: Fix committed
- 2013-10-30: Bluedroid tagged release 4.4 r0.9
- 2013-10-31: Android 4.4 released

Not fixed in 4.3

See also: Colin and John's talk on wednesday

Bluedroid on PC

- It's just C, why not?
- GDB: Single-stepping through packet parsing code
- In theory can be combined with HCI_USER_SOCKET

```
$ ps aux | grep bluedroid  
mikeryan 4168 0.2 0.0 60552 1480 pts/5 Sl+ 10:41 0:00 ./bluedroid
```


How Do You Pronounce “BlueZ”?

“That is the one single mystery. Nobody knows.”

- Marcel Holtmann, BlueZ maintainer

Thanks

- Marcel Holtmann
- BlueZ team
- Google

- CanSecWest
- iSEC Partners

Thank You

Mike Ryan

iSEC Partners

@mpeg4codec

mikeryan@isecpartners.com

<https://lacklustre.net/>