Outsmarting Bluetooth Smart

Mike Ryan
iSEC Partners
CanSecWest
Mar 14, 2014
Quick Note

Bluetooth Smart

Bluetooth Low Energy

BLE

all the same thing!
A Brief History of BLE

- 2010: Bluetooth 4.0
- 2011: BlackBerry 10
- 2012: iPhone 4S / iOS 5.0
- 2013: BLE explosion, crackle demonstrated
- 2014: Bluetooth 4.1, Android 4.3, BlackBerry 10
Simplified BLE Stack

apps → GATT  |  SM
         |  L2CAP
         |  Link Layer
         |  PHY

→ crackle

Ubertooth
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
<table>
<thead>
<tr>
<th>Service</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device Information</strong></td>
<td><strong>Manufacturer Name String</strong></td>
</tr>
<tr>
<td>Peripheral: Nike+ FuelBand</td>
<td>Data Type: string</td>
</tr>
<tr>
<td>Service: Device Information</td>
<td></td>
</tr>
<tr>
<td>UUID: 180A</td>
<td>ASCII: Nike</td>
</tr>
<tr>
<td>0x83CDC410-31DD-11E2-81C1-0800:</td>
<td>Hex: 0x4E696B65</td>
</tr>
<tr>
<td>UUID: 83CDC410-31DD-11E2-81C1-080</td>
<td>Decimal: 1701538126</td>
</tr>
<tr>
<td><strong>Model Number String</strong></td>
<td><strong>Date</strong></td>
</tr>
<tr>
<td>UUID: 2A24</td>
<td>2014/03/12 16:24:36:169</td>
</tr>
<tr>
<td><strong>Serial Number String</strong></td>
<td></td>
</tr>
<tr>
<td>UUID: 2A25</td>
<td></td>
</tr>
<tr>
<td><strong>Firmware Revision String</strong></td>
<td></td>
</tr>
<tr>
<td>UUID: 2A26</td>
<td></td>
</tr>
<tr>
<td><strong>Software Revision String</strong></td>
<td></td>
</tr>
<tr>
<td>UUID: 2A27</td>
<td></td>
</tr>
<tr>
<td><strong>Hardware Revision String</strong></td>
<td></td>
</tr>
</tbody>
</table>
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, #(_objc retainAutoreleasedReturnValue)
MOV     R7, R7
BLX     _objc_retainAutoreleasedReturnValue
MOV     R1, #(_OBJC_IVAR_$_BluetoothManager._DataStreamCharUUID - 0xD2F0A) ; CBUUID
ADD     R1, PC ; CBUUID *xxxxxxDataStreamCharUUID
```

```
MOV     R1, R5
ADD     R2, PC ; "0x0979"
BLX     _OBJC_CLASS_$_CBUUID
MOV     R7, R7
BLX     _objc_retainAutoreleasedReturnValue
MOV     R1, #(_OBJC_IVAR_$_BluetoothManager._DataStreamCharUUID - 0xD2F0A) ; CBUUID
ADD     R1, PC ; CBUUID *xxxxxxDataStreamCharUUID
LDR     R6, [R1] ; CBUUID *xxxxxxDataStreamCharUUID;
```

“8xxxxxxx-xxx2-4FE5-AA46-111111111111”
“0x0979”
CBUUID *xxxxxxDataStreamCharUUID
Clone the Device

→ BLE devices are role-flexible
→ This includes BlueZ
Length Fields Aplenty

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

```
06 09 05 00 04 00 0c 20 00 2c 00
data
```
### HCI H4

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>0.000273</td>
<td></td>
<td></td>
<td>HCI_CMD</td>
<td>11</td>
<td>Sent Delete Stored Link Key</td>
</tr>
<tr>
<td>22</td>
<td>0.000695</td>
<td></td>
<td></td>
<td>HCI_EVT</td>
<td>9</td>
<td>Rcvd Command Complete (Delete Stored Link Key)</td>
</tr>
<tr>
<td>23</td>
<td>0.000219</td>
<td></td>
<td></td>
<td>HCI_CMD</td>
<td>4</td>
<td>Sent LE Read Buffer Size</td>
</tr>
<tr>
<td>24</td>
<td>0.000755</td>
<td></td>
<td></td>
<td>HCI_EVT</td>
<td>10</td>
<td>Rcvd Command Complete (LE Read Buffer Size)</td>
</tr>
<tr>
<td>25</td>
<td>0.000184</td>
<td></td>
<td></td>
<td>HCI_CMD</td>
<td>4</td>
<td>Sent LE Read Advertising Channel Tx Power</td>
</tr>
</tbody>
</table>
| 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  | 7      | 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) |

**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)
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
→ Very good code!

→ Mutative fuzzing

```c
uint 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);

    return SUCCESS;
}
```

gatttool scripting ftw!
## Victims Targets

- Smartphones
  - Android: Bluedroid
  - iOS
  - Windows Phone 8.1
  - BlackBerry 10

- Devices
  - TI
  - Nordic
Generic BLE Device

- Sensor
- Output
- BLE SoC

Diagram showing the components of a Generic BLE Device.
BLE System-on-Chip

- CPU core
  - 8051 or Cortex-M0
- BLE radio
- Flash
Primary Target: Bluedroid

```bash
~/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
  → DeMo
    → DeMo
      → DeMo
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);
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

```bash
$ 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/