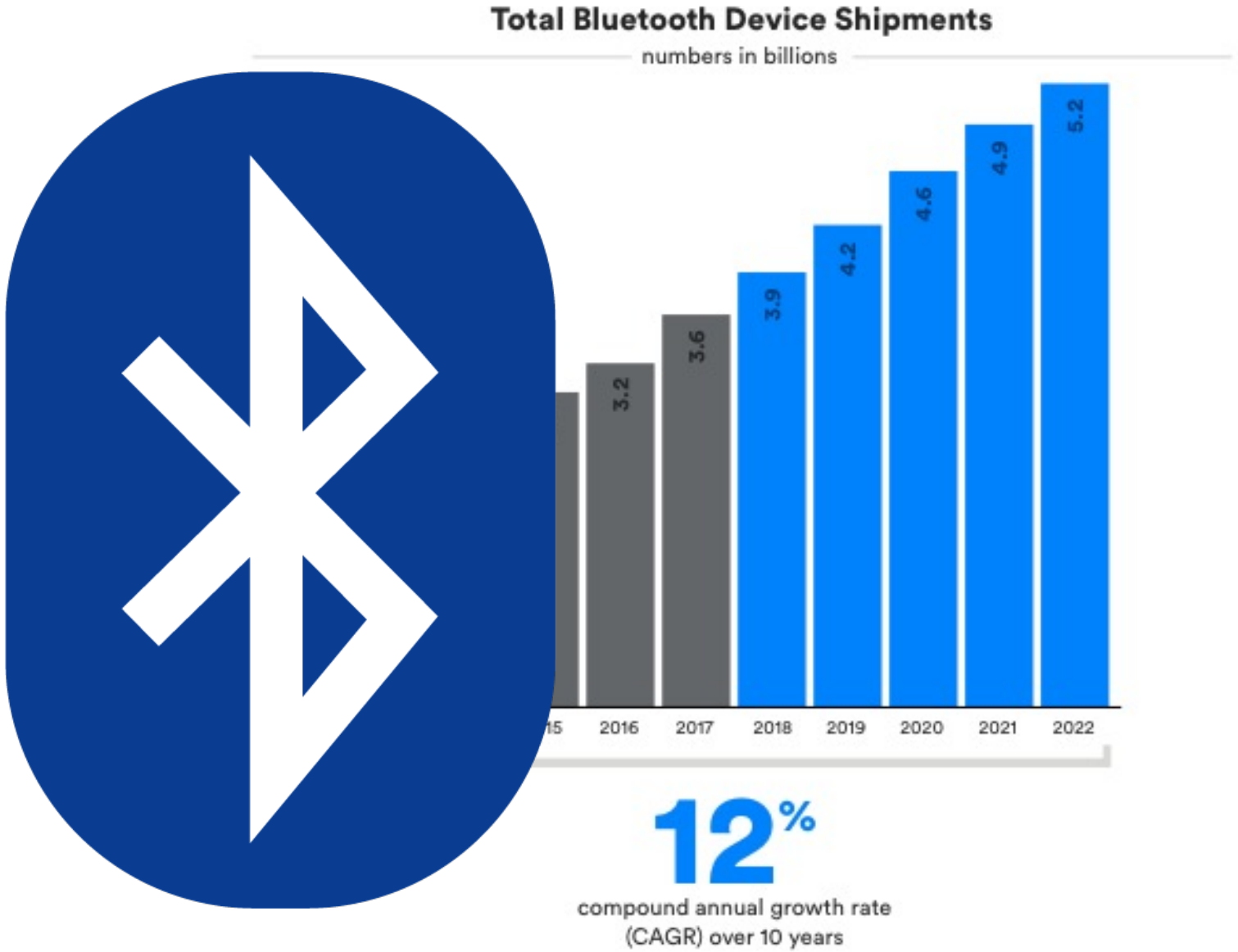
Bluetooth Hacking: Tools and Techniques

Mike Ryan

Founder
ICE9 Consulting
@mpeg4codec



ICE9 CONSULTING


gifbin.com

Who is this talk for?

Bluetooth device developers

Penetration testers

Managers

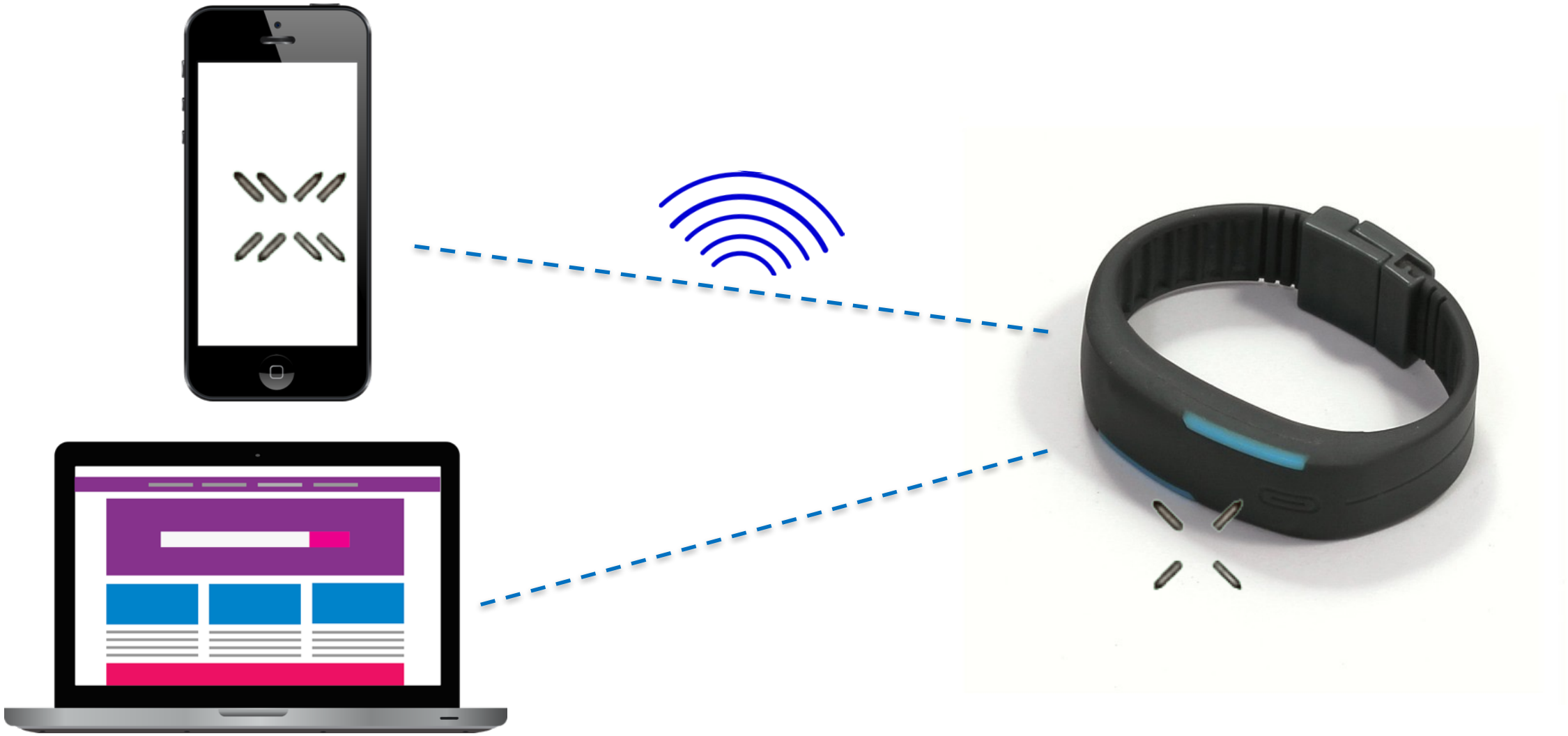
Structure of the Talk

Overview of Bluetooth

RE Process and Tools

Case Studies

Bluetooth and the Reverse Engineering Process



ICE9
CONSULTING

Reverse Engineering Process

1. Do something with the device and app
2. Capture the data sent via Bluetooth
3. Analyze

???

Sniffing Bluetooth is Hard

Pros: 100% reliable

Cons: $20,000

ICE9 CONSULTING

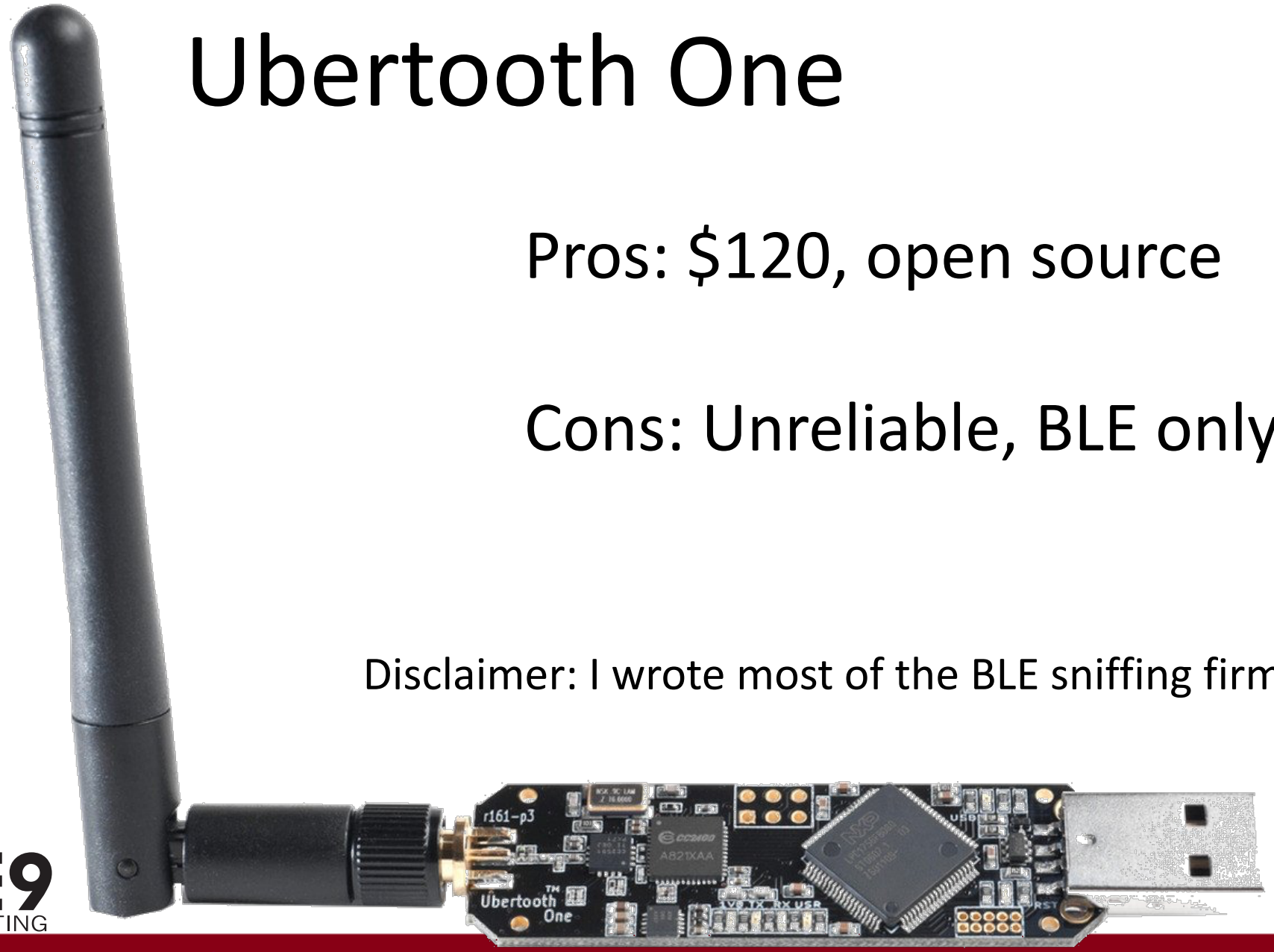
Disclaimer: I have never used one

Ubertooth One

Pros: $120, open source

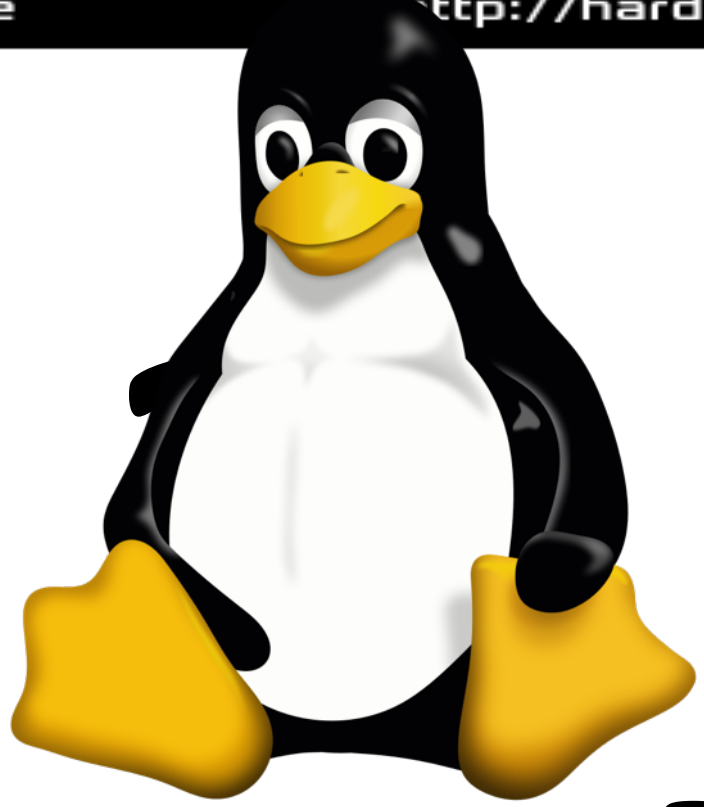
Cons: Unreliable, BLE only

Disclaimer: I wrote most of the BLE sniffing firmware



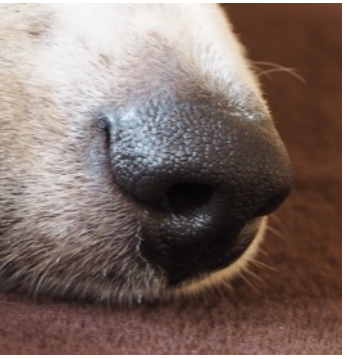
ICE9 CONSULTING

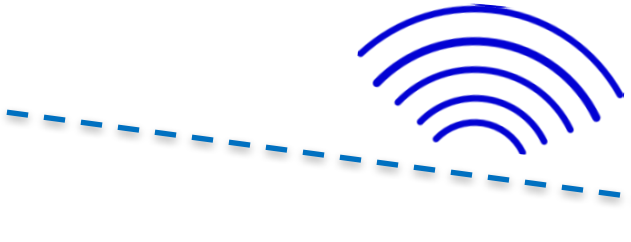
re is



Be OS X lay!

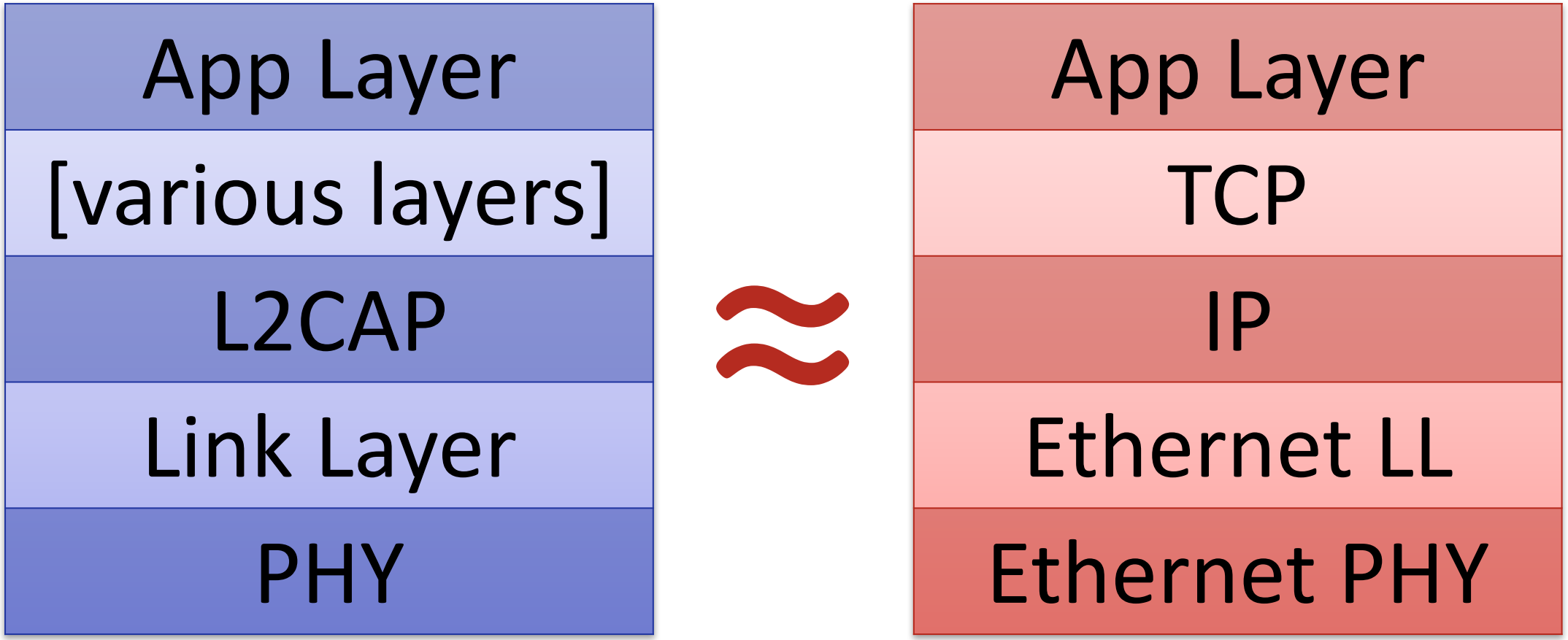
ICE9 CONSULTING



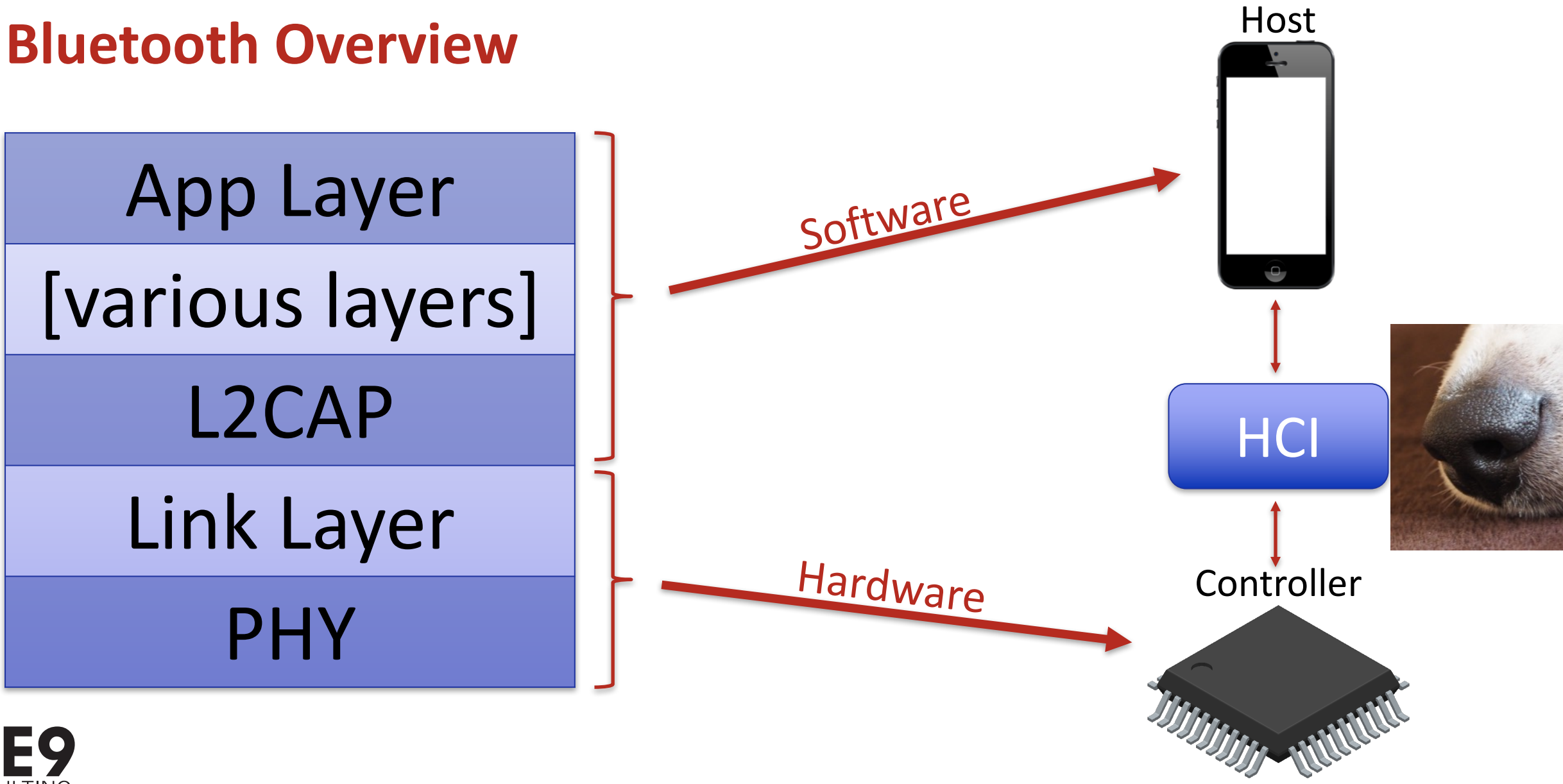


ICE9 CONSULTING

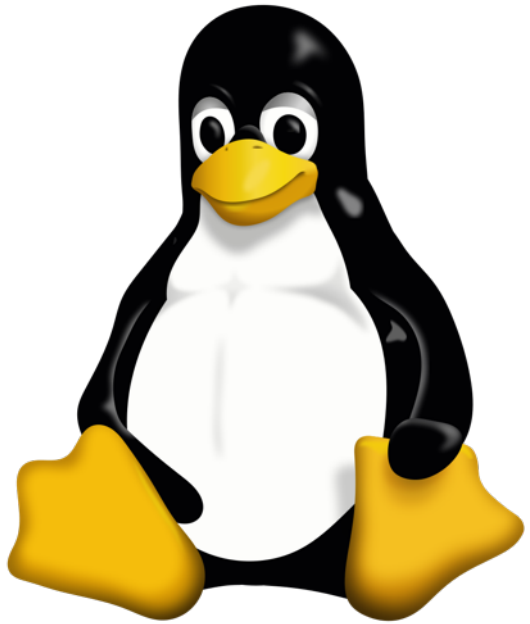
Bluetooth Overview



Bluetooth Overview

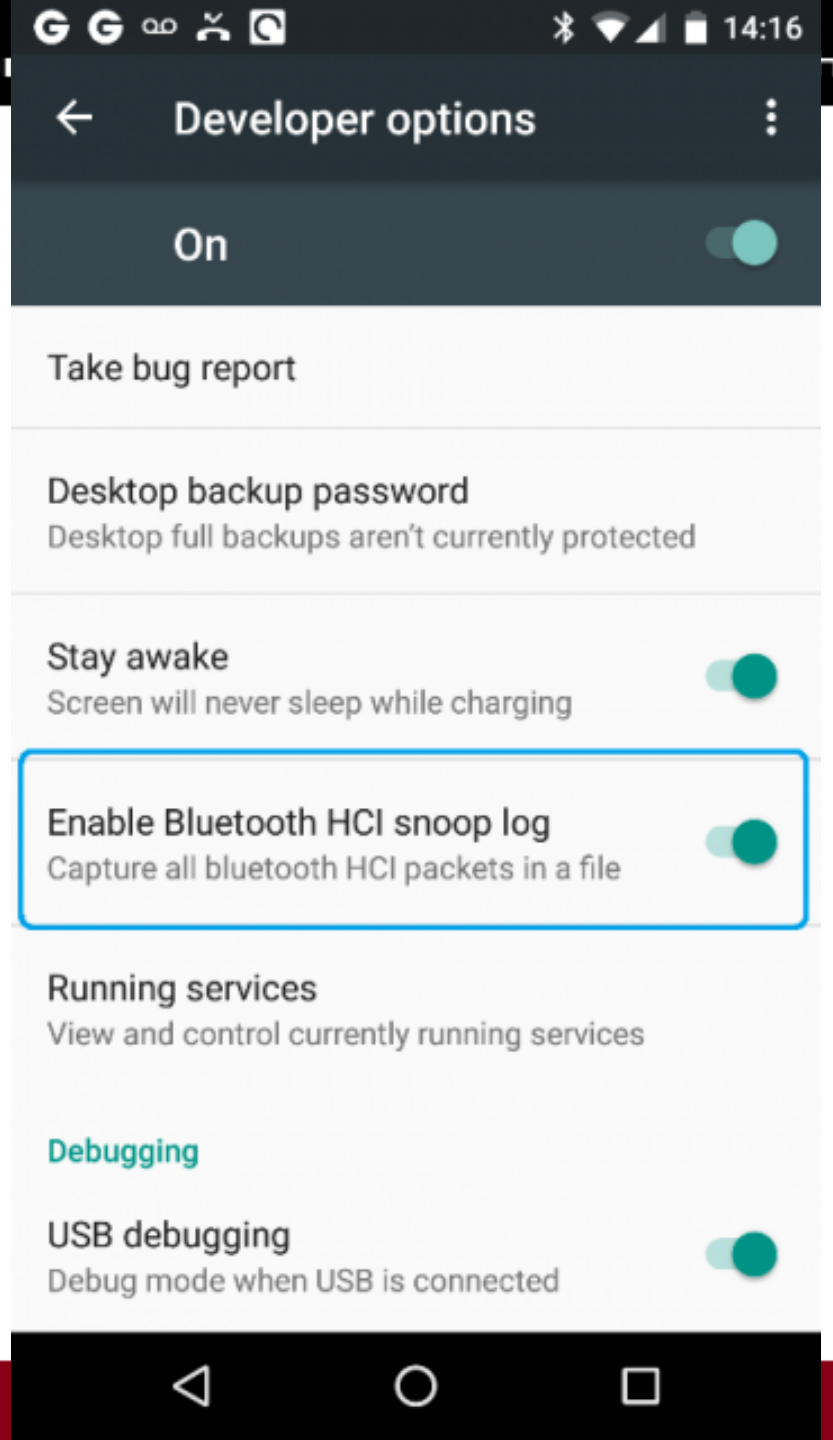


Logging HCI



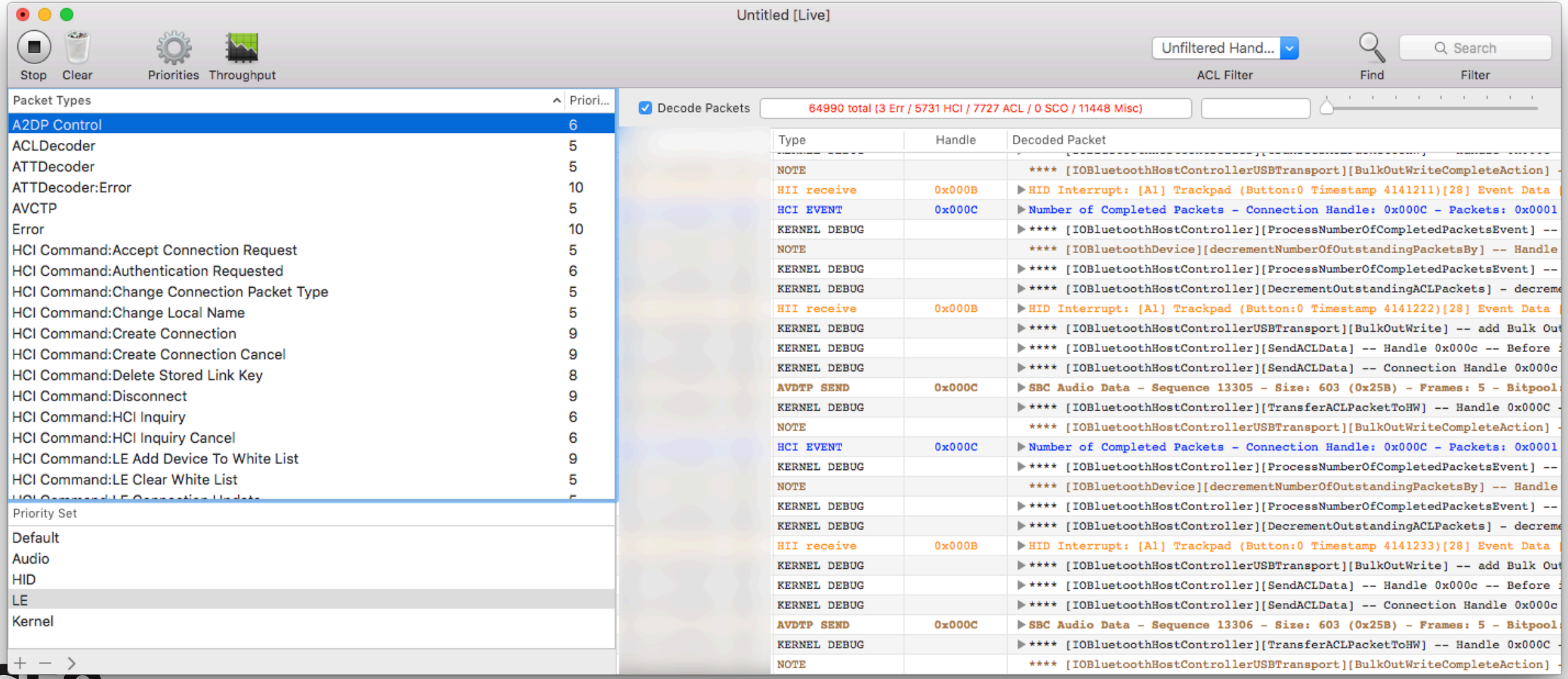
OSX

ICE9 CONSULTING



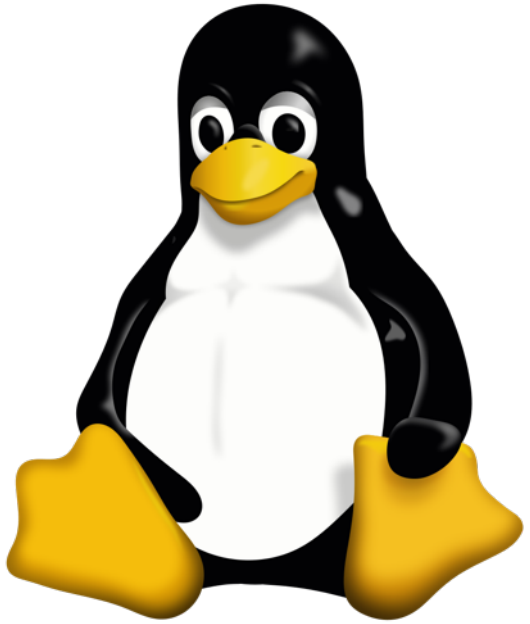
ICE9 CONSULTING

OS X packet logger



ICE9 CONSULTING

Linux Logging



```
$ sudo btmon -w logfile.log
```

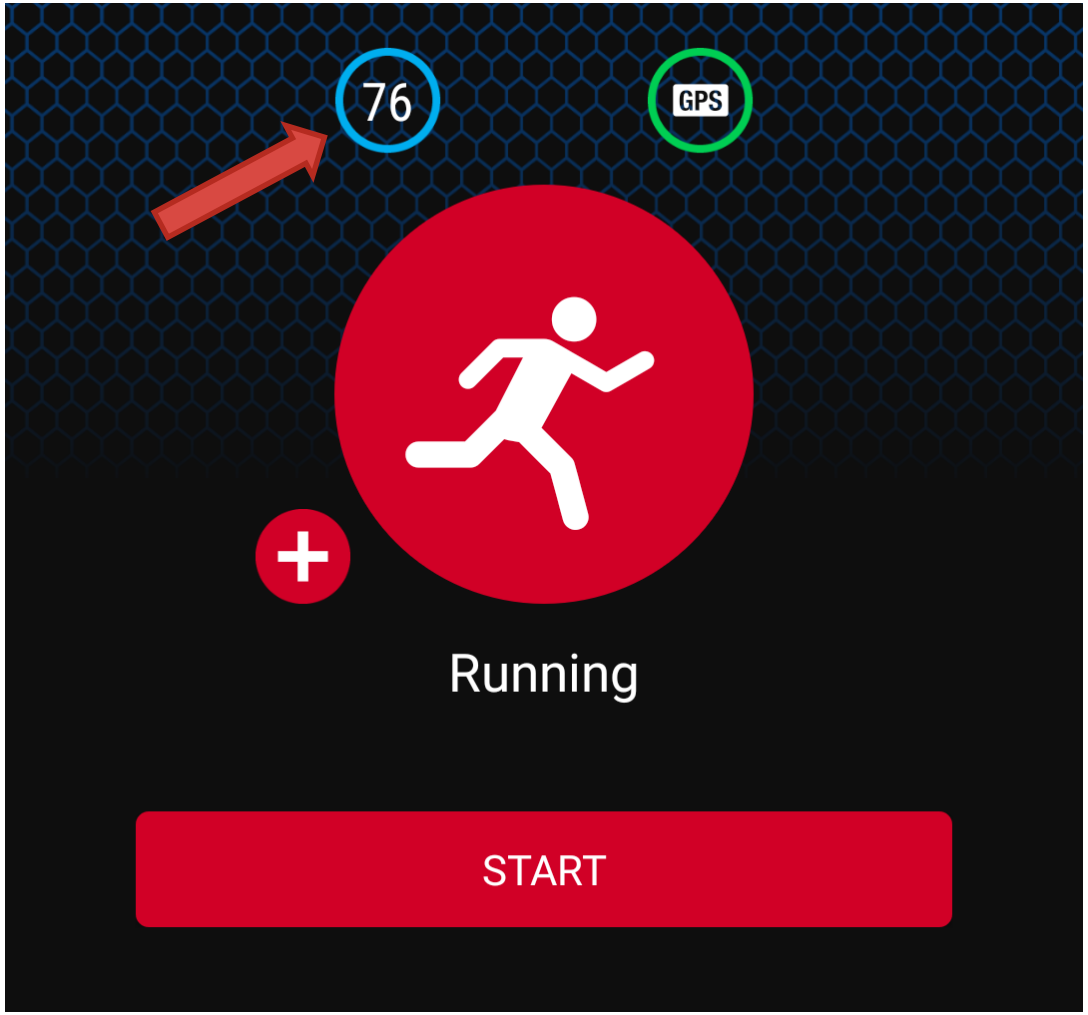
Reverse Engineering Process

1. Do something with the device and app
2. Capture the data sent via Bluetooth
3. Analyze

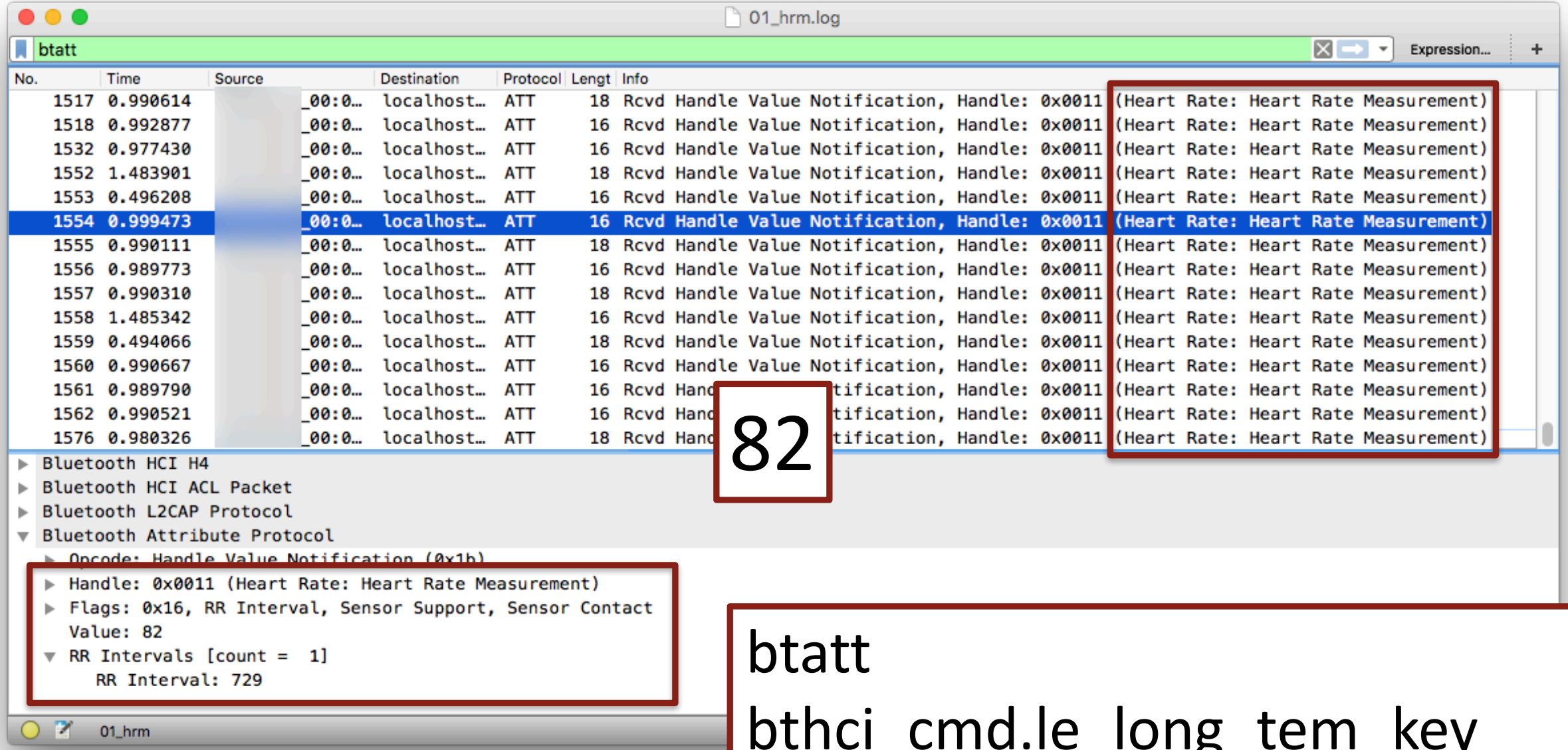
Case Studies

ICE9 CONSULTING

Case Study 1: BLE Heart Rate Monitor



ICE9 CONSULTING



btatt

bthci_cmd.le_long_tem_key

No encryption

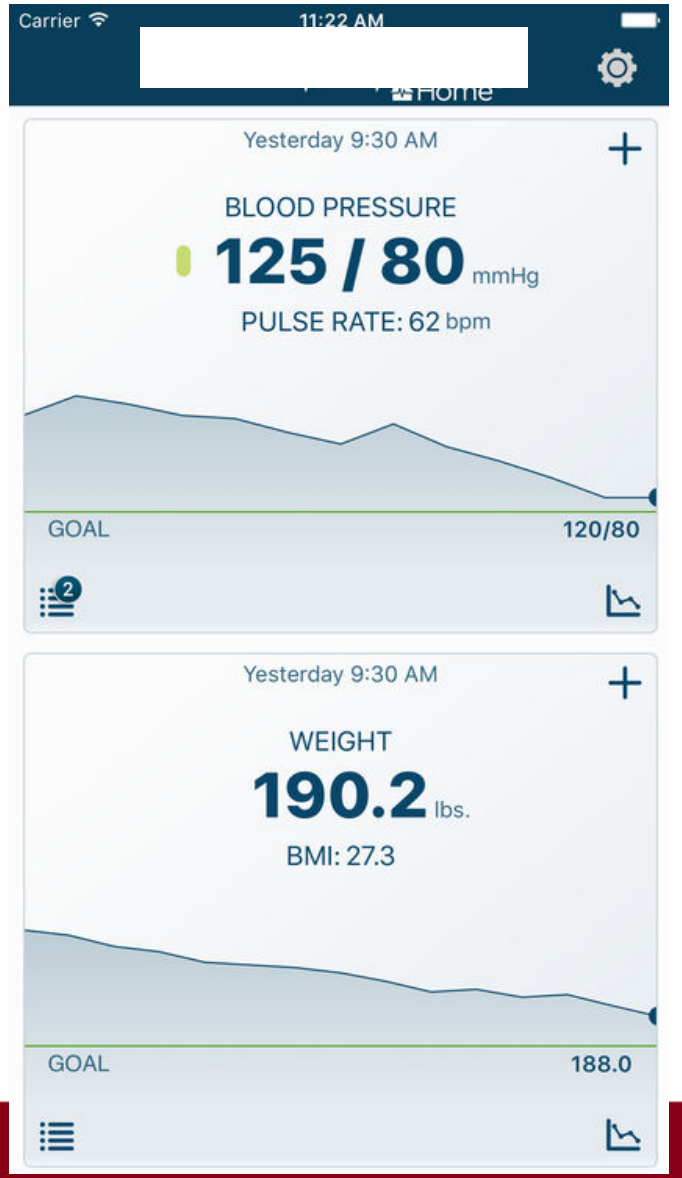
Case Study 1: Conclusions

- Wireshark is incredible
- Getting Bluetooth logs is practical

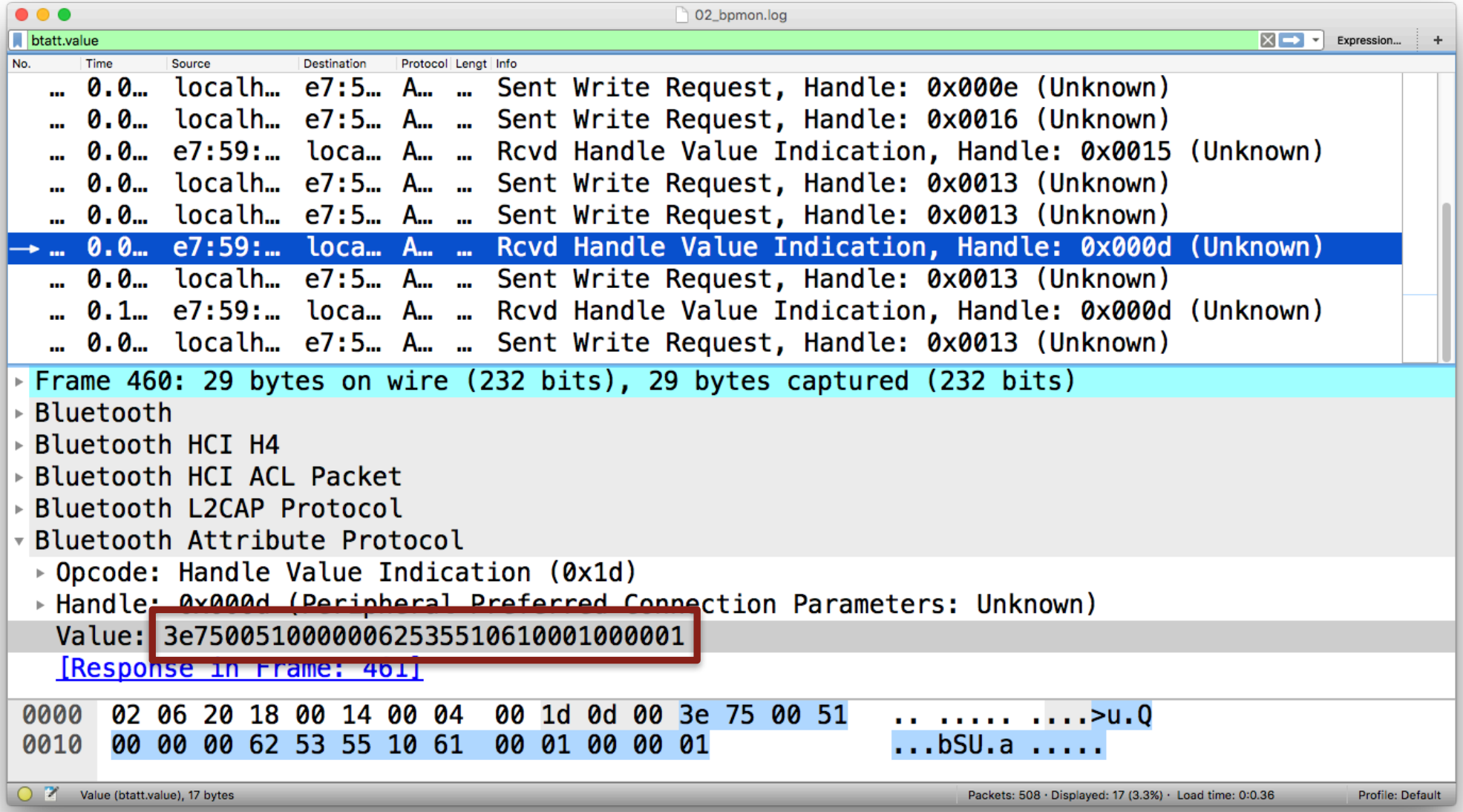
```
# gatttool -b 00:[redacted]:7E --primary
attr handle = 0x0001, end grp handle = 0x000b
    uuid: 00001800-0000-1000-8000-00805f9b34fb
attr handle = 0x000c, end grp handle = 0x000e
    uuid: 00001801-0000-1000-8000-00805f9b34fb
attr handle = 0x000f, end grp handle = 0x0014
    uuid: 0000180d-0000-1000-8000-00805f9b34fb
attr handle = 0x0015, end grp handle = 0x0023
    uuid: 0000180a-0000-1000-8000-00805f9b34fb
attr handle = 0x0024, end grp handle = 0x0026
    uuid: 0000180f-0000-1000-8000-00805f9b34fb
attr handle = 0x0027, end grp handle = 0xffff
    uuid: 6217ff49-ac7b-547e-eecf-016a06970ba9
```

ICE9 CONSULTING

Case Study 2: BLE Blood Pressure Monitor



ICE9 CONSULTING



ICE9 CONSULTING

GATT Services | Bluet

https://www.bluetooth.com/specifications/gatt/services

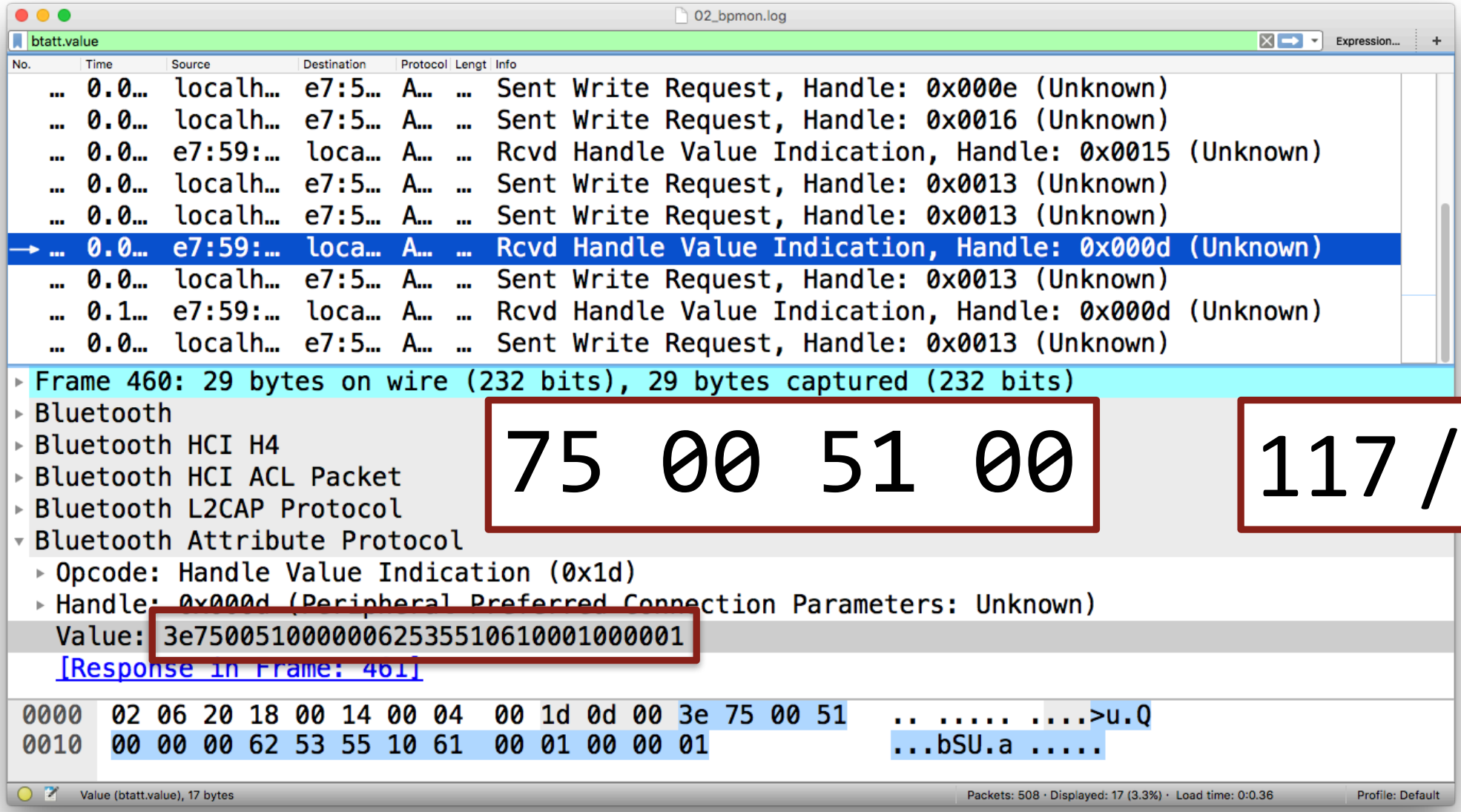
- Protocol Specifications
- **GATT Specifications**
  - GATT Overview
  - GATT Characteristics
  - GATT Declarations
  - GATT Descriptors
  - GATT Services
  - Mesh GATT Services XML
  - Available Schemas
- Errata Service Releases
- Qualification Test Requirements
- Assigned Numbers
- Specifications in Development
- Interoperable Prototype Test Events (IOP)

Name	Uniform Type Identifier	Assigned Number	Specification
Generic Access	org.bluetooth.service.generic_access		
Alert Notification Service	org.bluetooth.service.alert_notification		
Automation IO	org.bluetooth.service.automation_io		
Battery Service	org.bluetooth.service.battery_service		
Blood Pressure	org.bluetooth.service.blood_pressure		
Body Composition	org.bluetooth.service.body_composition		
Bond Management Service	org.bluetooth.service.bond_management		
Continuous Glucose Monitoring	org.bluetooth.service.continuous_glucose		
Current Time Service	org.bluetooth.service.current_time		
Cycling Power	org.bluetooth.service.cycling_power		
Cycling Speed and Cadence	org.bluetooth.service.cycling_speed_and		
Device Information	org.bluetooth.service.device_information	0x180A	GSS
HTTP Proxy	org.bluetooth.service.http_proxy	0x1823	GSS

Write characteristic: Phone → Device

Notify characteristic: Device → Phone

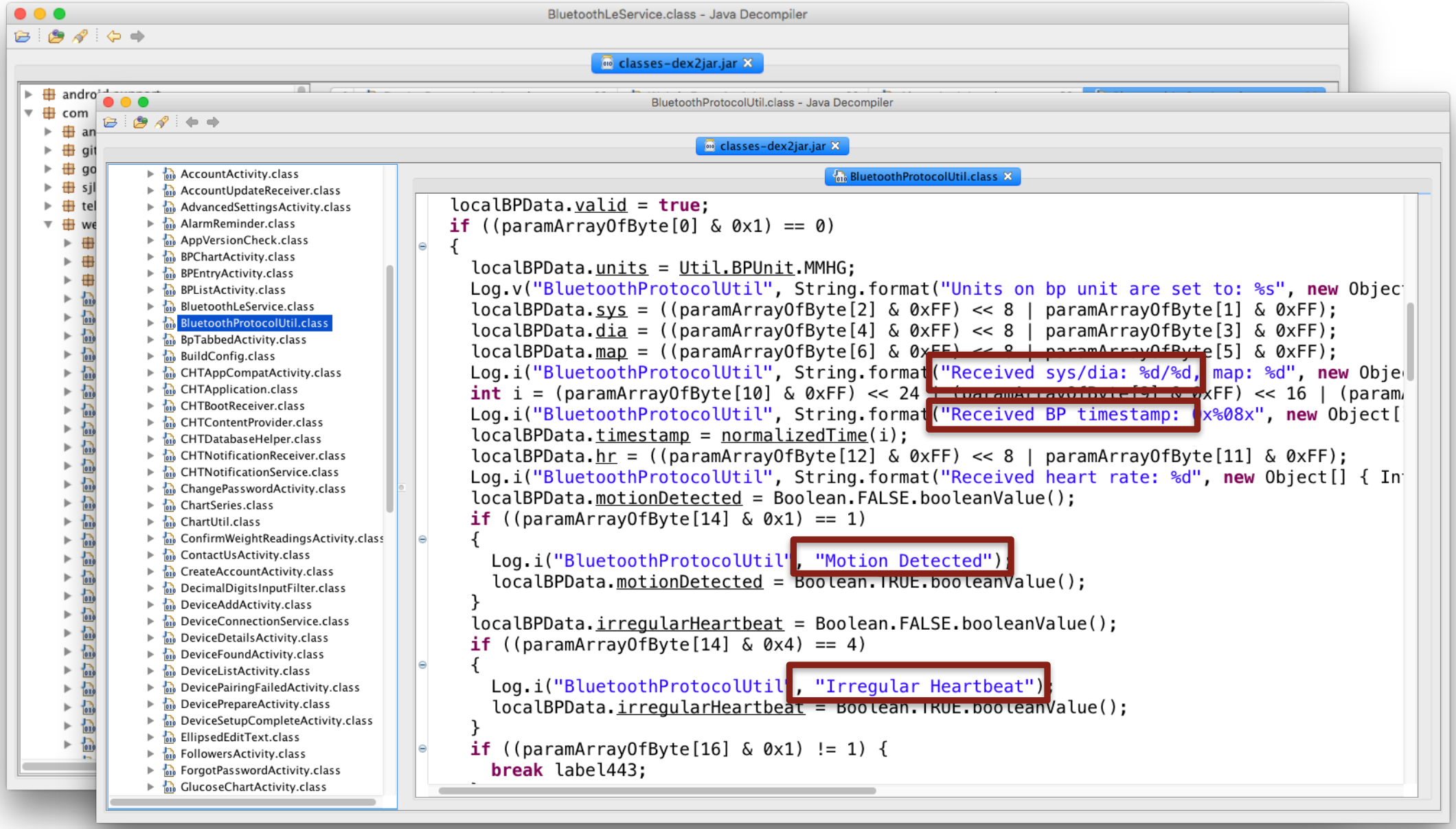
ICE9 CONSULTING



75 00 51 00

117 / 81

ICE9 CONSULTING



ICE9 CONSULTING

Conclusions: BP Monitor

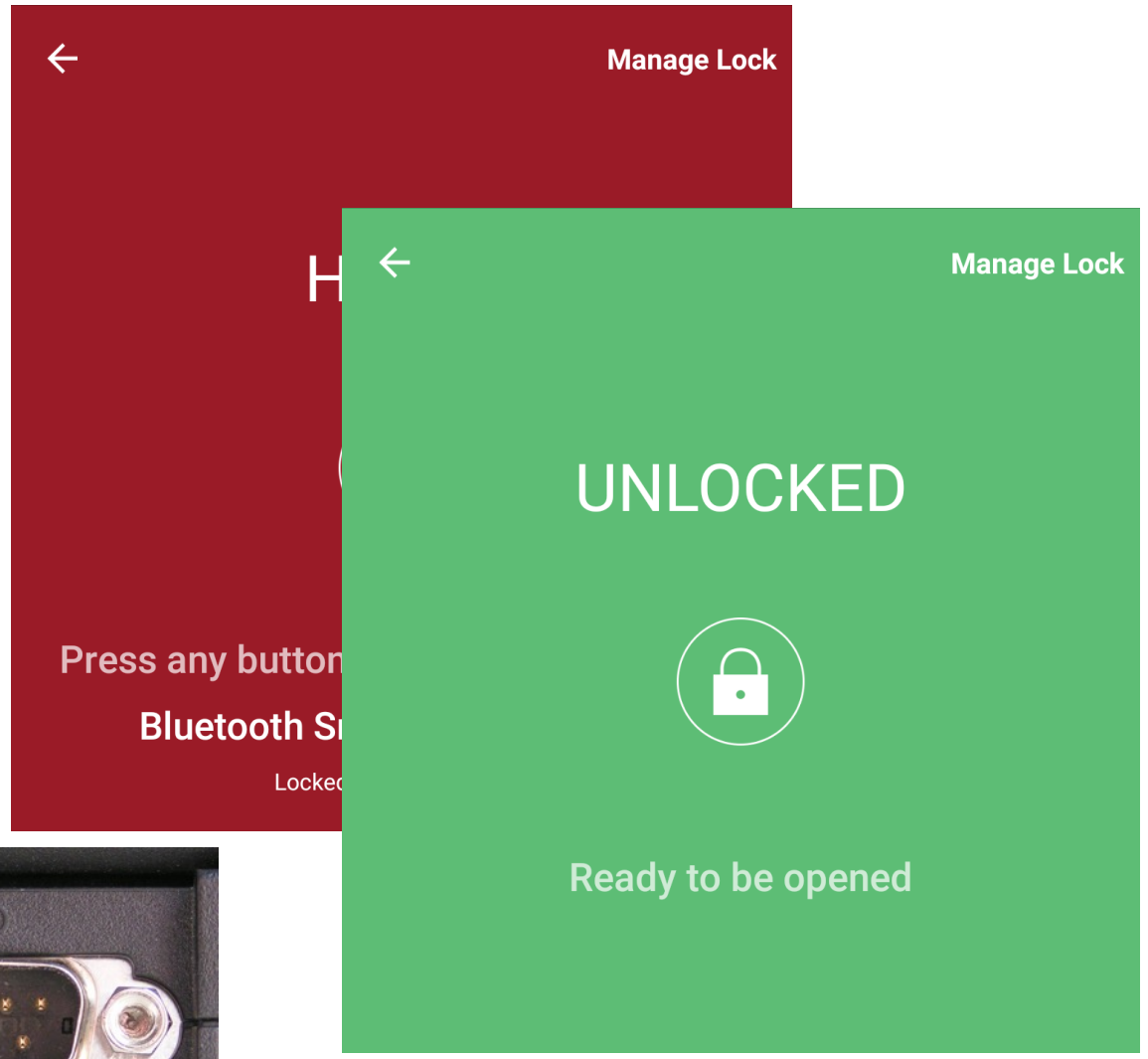
- “Hidden” serial port
- Normal binary protocol reverse engineering
- Look in the app

Once again – No Encryption

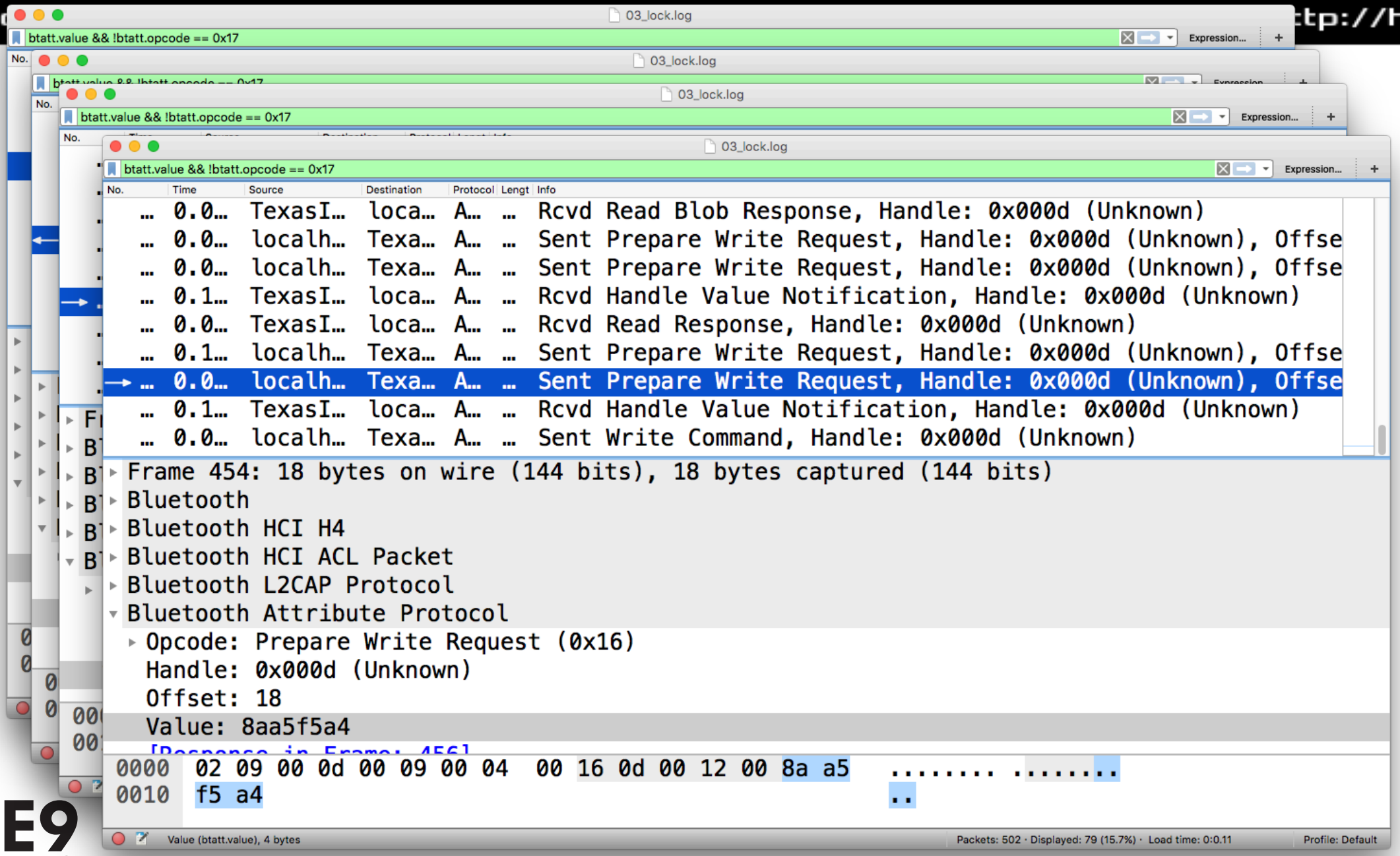
Firmware Update Service

ICE9 CONSULTING

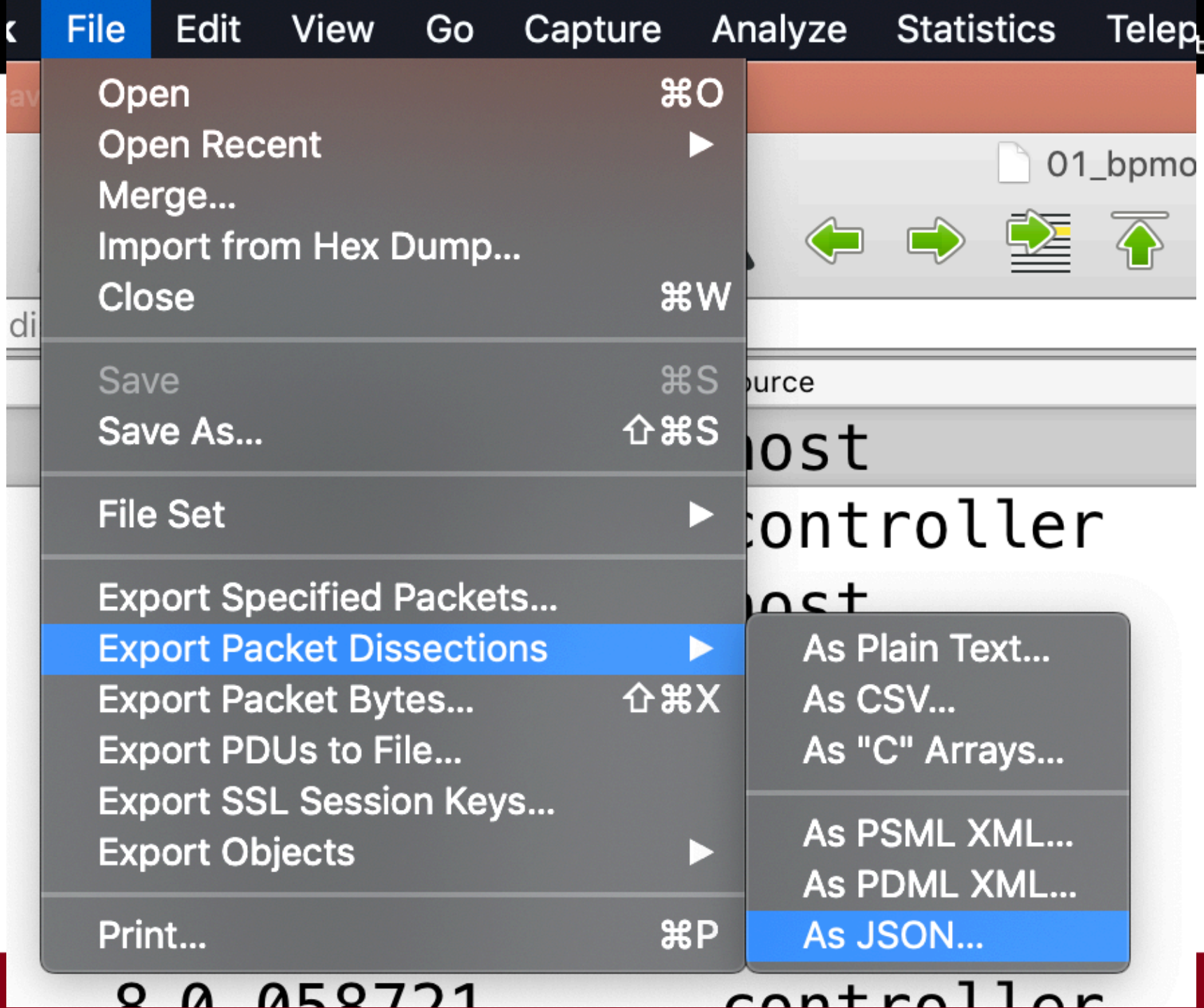
Case Study 3: BLE Padlock

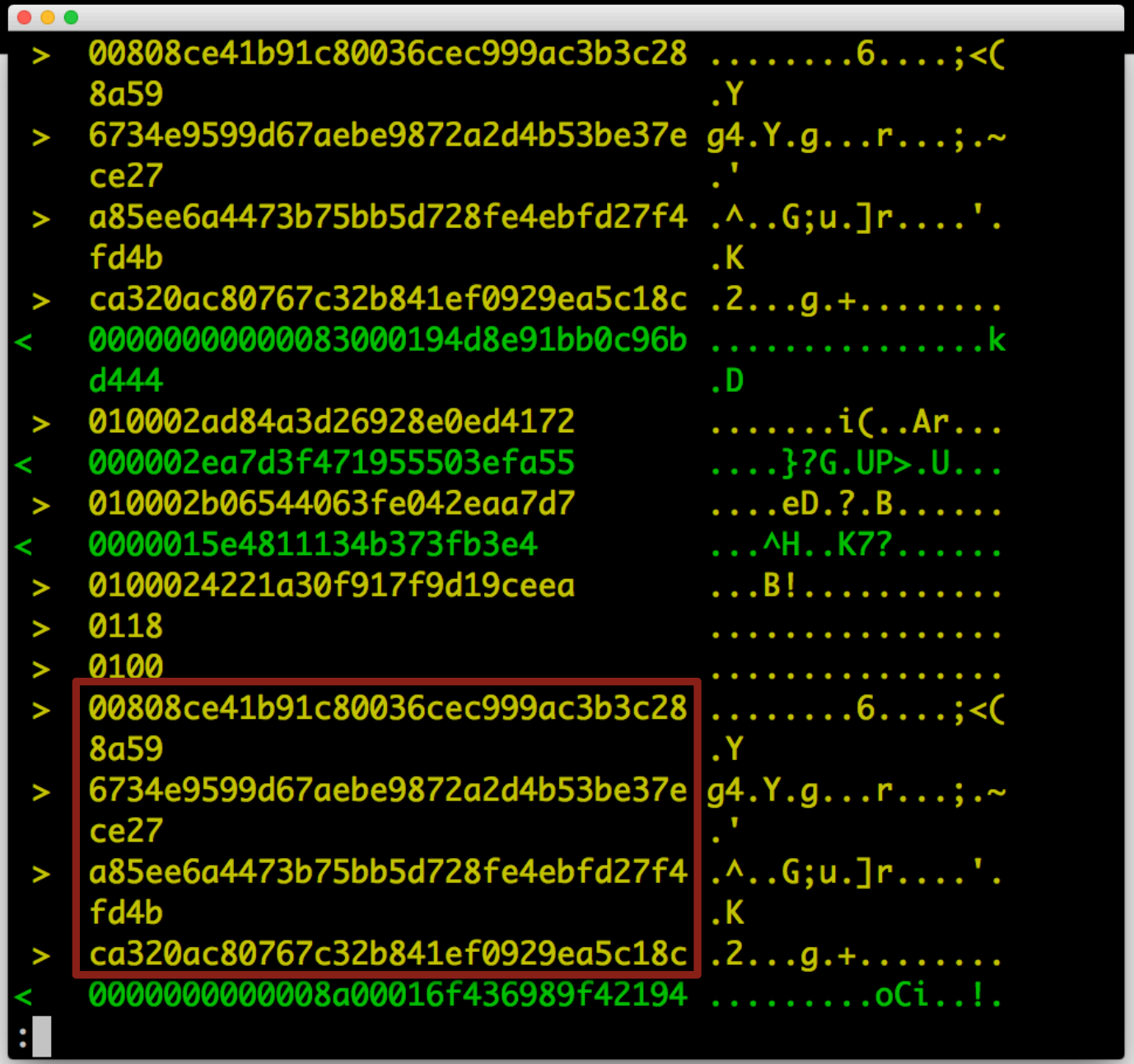


ICE9 CONSULTING

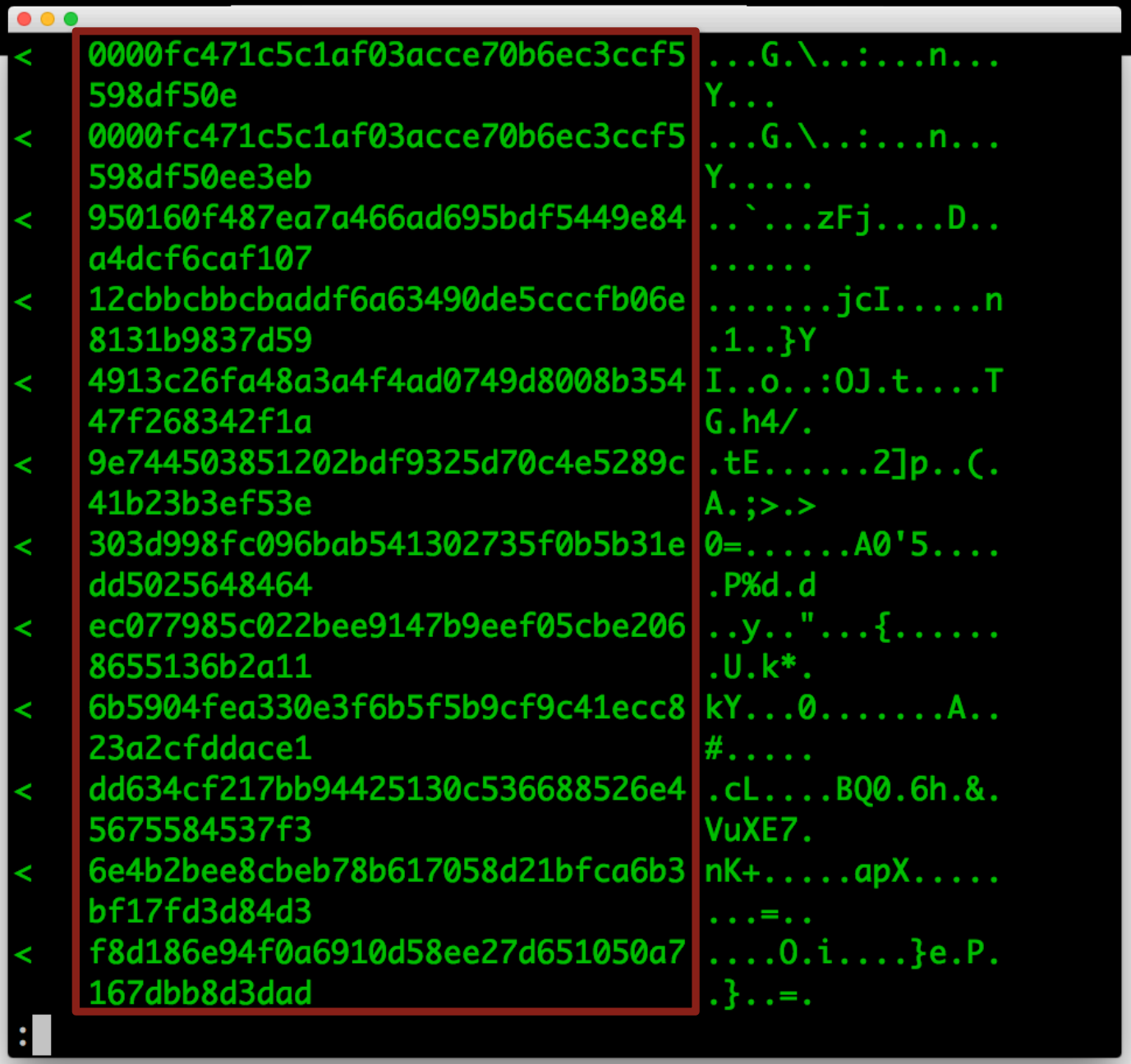


ICE9 CONSULTING


File
Edit
View
Go
Capture
Analyze
Statistics
Open ⌘O
Open Recent
Merge...
Import from Hex Dump...
Close ⌘W
Save ⌘S
Save As... ⇧⌘S
File Set
Export Specified Packets...
Export Packet Dissections
Export Packet Bytes... ⇧⌘X
Export PDUs to File...
Export SSL Session Keys...
Export Objects
Print... ⌘P
As Plain Text...
As CSV...
As "C" Arrays...
As PSML XML...
As PDML XML...
As JSON...



ICE9 CONSULTING



ICE9 CONSULTING

Conclusions: Padlock

- Developers were security-minded
- Home-grown crypto is fraught with peril

Intermission

Case Study 4: Classic Bluetooth Headset

~~A2DP? HFP?~~

Reasons

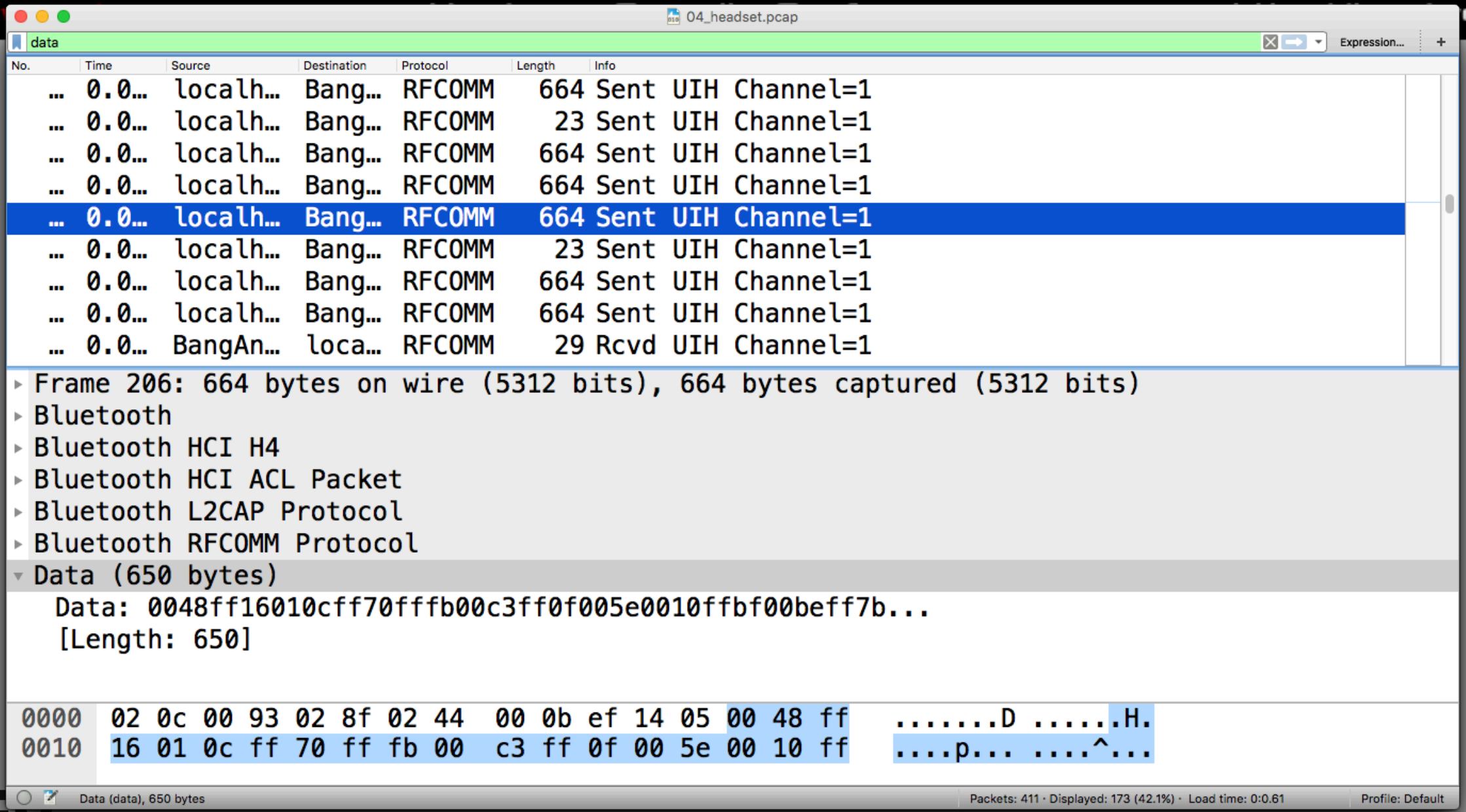
29 Proximity Profile (PXP)

30 Serial Port Profile (SPP)

31 Service Discovery Application Pr

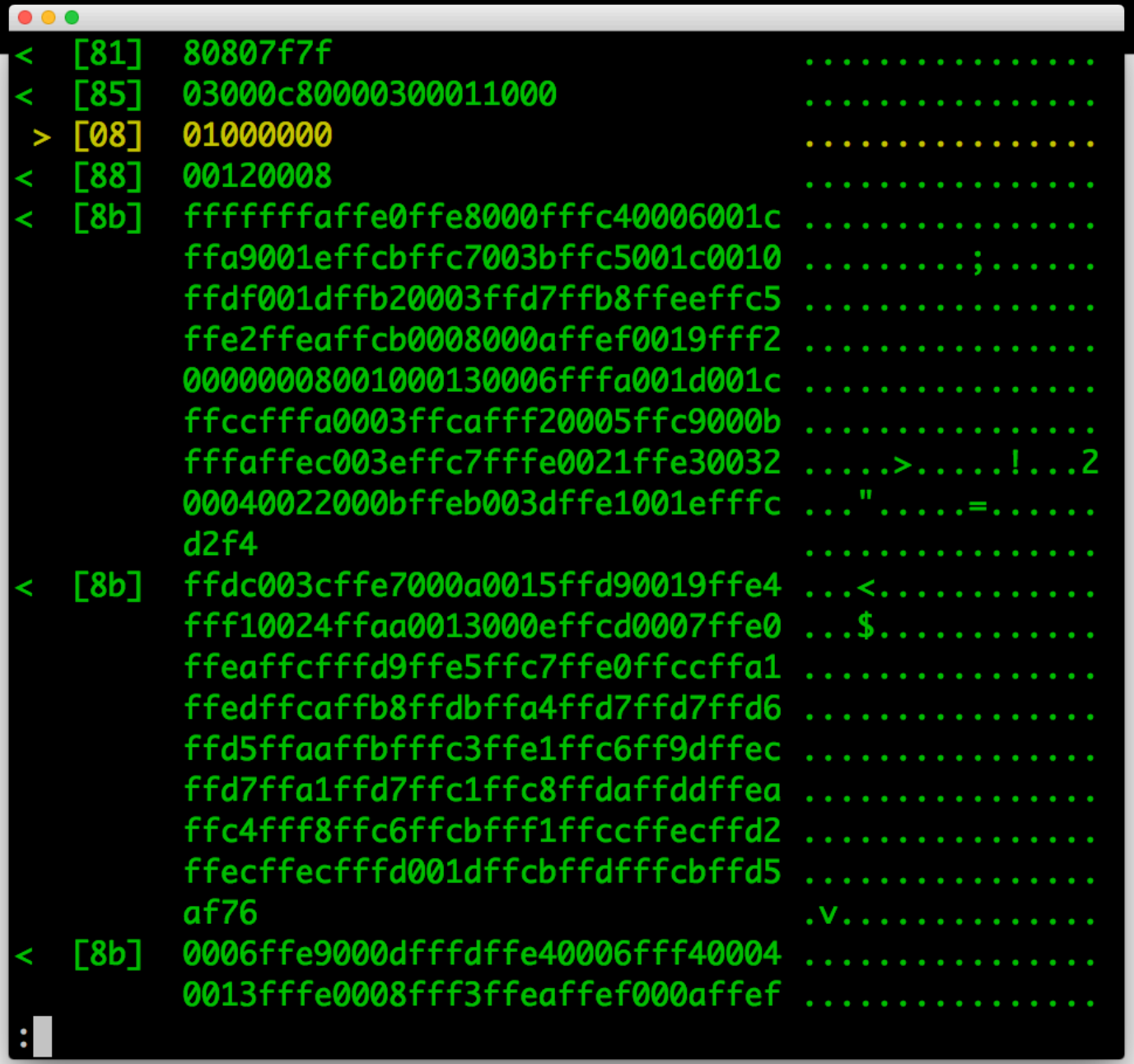
32 SIM Access Profile (SAP, SIM, r

ICE9 CONSULTING


04_headset.pcap
data
Expression...
No. Time Source Destination Protocol Length Info
… 0.0… localh… Bang… RFCOMM 664 Sent UIH Channel=1
… 0.0… localh… Bang… RFCOMM 23 Sent UIH Channel=1
… 0.0… localh… Bang… RFCOMM 664 Sent UIH Channel=1
… 0.0… localh… Bang… RFCOMM 664 Sent UIH Channel=1
… 0.0… localh… Bang… RFCOMM 664 Sent UIH Channel=1
… 0.0… localh… Bang… RFCOMM 23 Sent UIH Channel=1
… 0.0… localh… Bang… RFCOMM 664 Sent UIH Channel=1
… 0.0… localh… Bang… RFCOMM 664 Sent UIH Channel=1
… 0.0… BangAn… loca… RFCOMM 29 Rcvd UIH Channel=1
Frame 206: 664 bytes on wire (5312 bits), 664 bytes captured (5312 bits)
Bluetooth
Bluetooth HCI H4
Bluetooth HCI ACL Packet
Bluetooth L2CAP Protocol
Bluetooth RFCOMM Protocol
Data (650 bytes)
Data: 0048ff16010cff70fffb00c3ff0f005e0010ffbf00beff7b...
[Length: 650]
0000 02 0c 00 93 02 8f 02 44 00 0b ef 14 05 00 48 ff .....D .....H.
0010 16 01 0c ff 70 ff fb 00 c3 ff 0f 00 5e 00 10 ff ....p... ....^...
Data (data), 650 bytes
Packets: 411 · Displayed: 173 (42.1%) · Load time: 0:0.61
Profile: Default

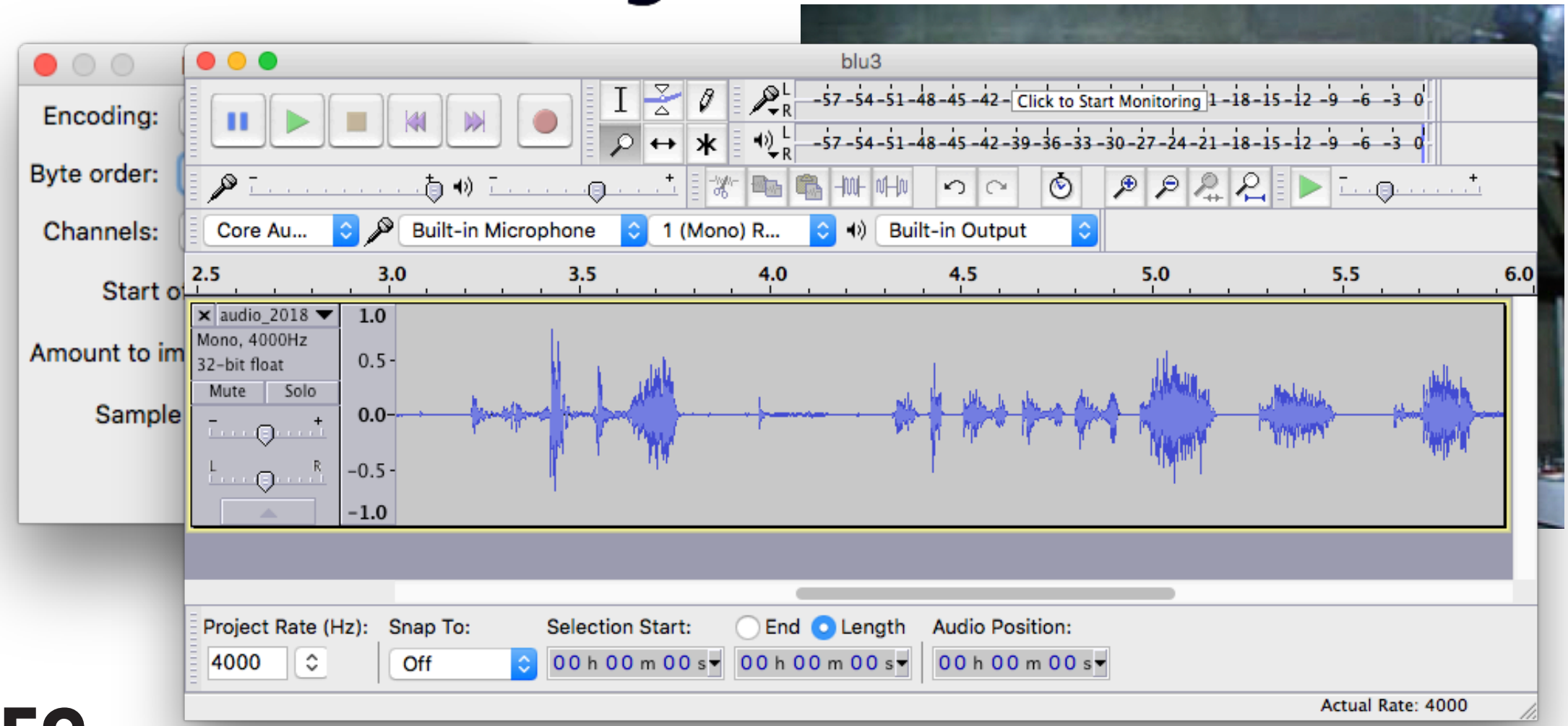
80 77 DD AF	magic number
XX	opcode
YY	sequence number
ZZ ZZ	length (16 bit little endian)
...	data
CC CC	checksum

ICE9 CONSULTING



```
<  [81]  80807f7f                                 ................
<  [85]  03000c80000300011000                     ................
 >  [08]  01000000                                 ................
<  [88]  00120008                                 ................
<  [8b]  fffffffaffe0ffe8000fffc40006001c         ................
        ffa9001effcbffc7003bffc5001c0010         .........;......
        ffdf001dffb20003ffd7ffb8ffeeffc5         ................
        ffe2ffeaffcb0008000affef0019fff2         ................
        00000008001000130006fffa001d001c         ................
        ffccfffa0003ffcafff20005ffc9000b         ................
        fffaffec003effc7fffe0021ffe30032         .....>.....!...2
        00040022000bffeb003dffe1001efffc         ..."......=.....
        d2f4                                     ................
<  [8b]  ffdc003cffe7000a0015ffd90019ffe4         ...<............
        fff10024ffaa0013000effcd0007ffe0         ...$............
        ffeaffcfffd9ffe5ffc7ffe0ffccffa1         ................
        ffedffcaffb8ffdbffa4ffd7ffd7ffd6         ................
        ffd5ffaaffbfffc3ffe1ffc6ff9dffec         ................
        ffd7ffa1ffd7ffc1ffc8ffdaffddffea         ................
        ffc4fff8ffc6ffcbfff1ffccffecffd2         ................
        ffecffecfffd001dffcbffdfffcbffd5         ................
        af76                                     .v..............
<  [8b]  0006ffe9000dfffdffe40006fff40004         ................
        0013fffe0008fff3ffeaffef000affef         ................
:
```

Audacity®



ICE9 CONSULTING

Conclusions: Headset

- The techniques apply equally well to BR and BLE
- Ultimately boils down to basic RE

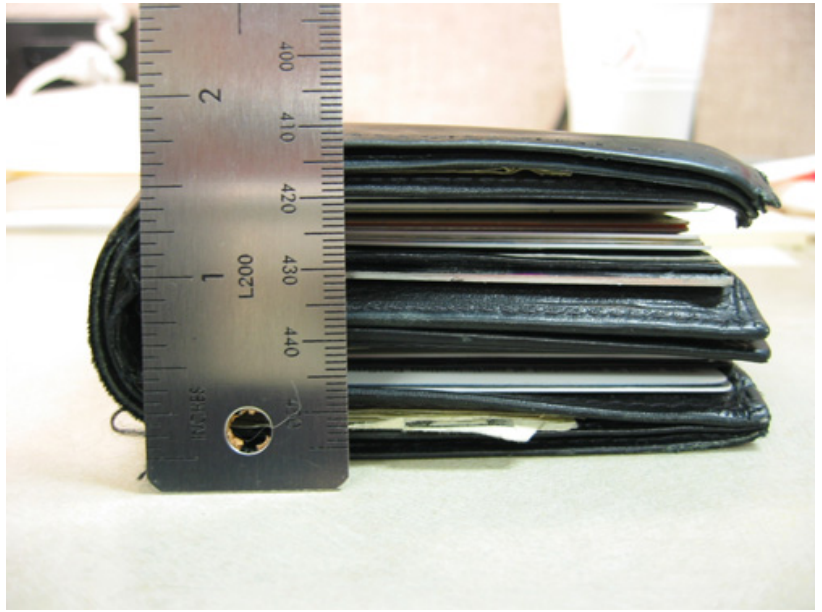
Case Study 5: BLE Credit Card

ICE9 CONSULTING

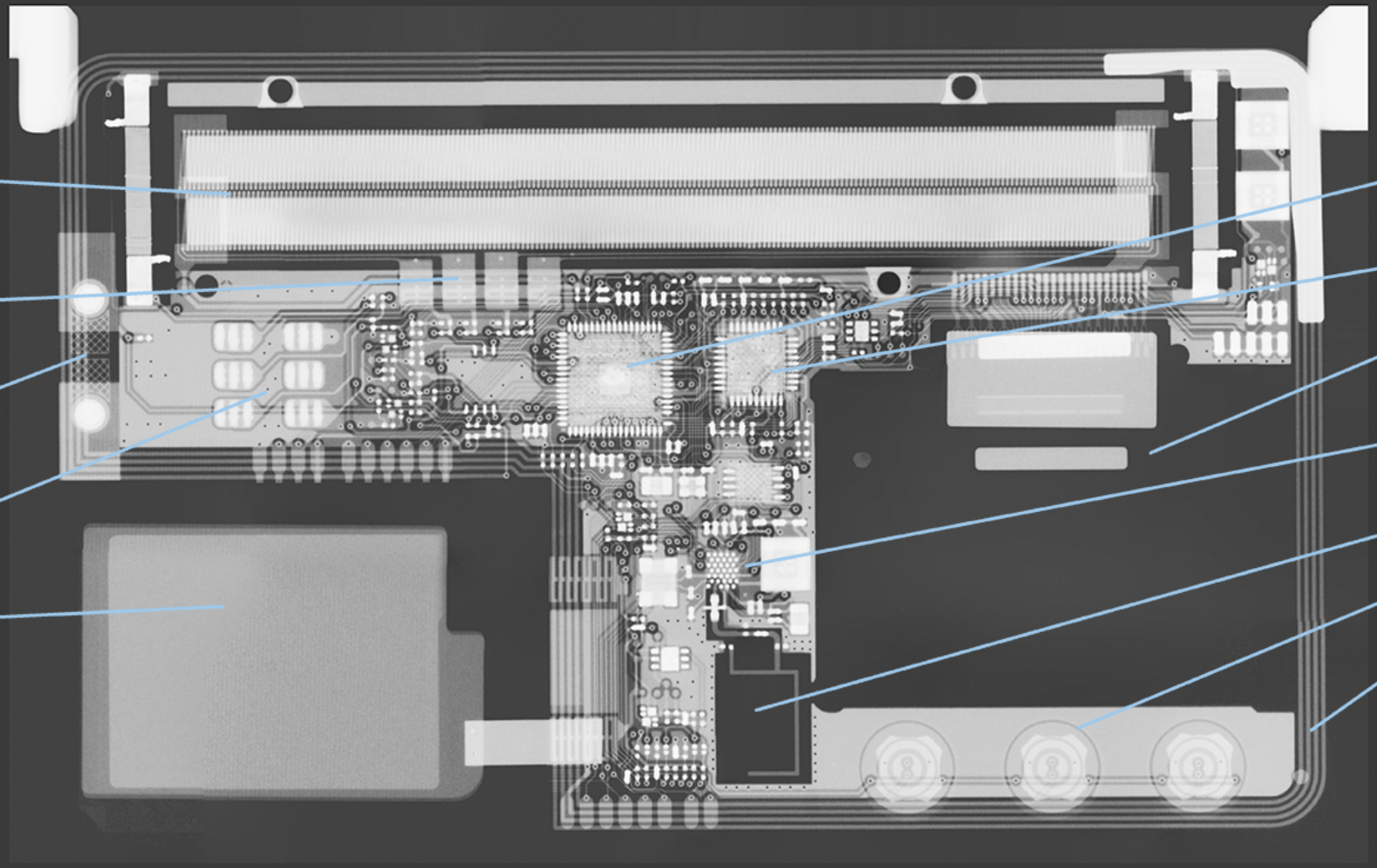
What is a Bluetooth credit card?

+

==

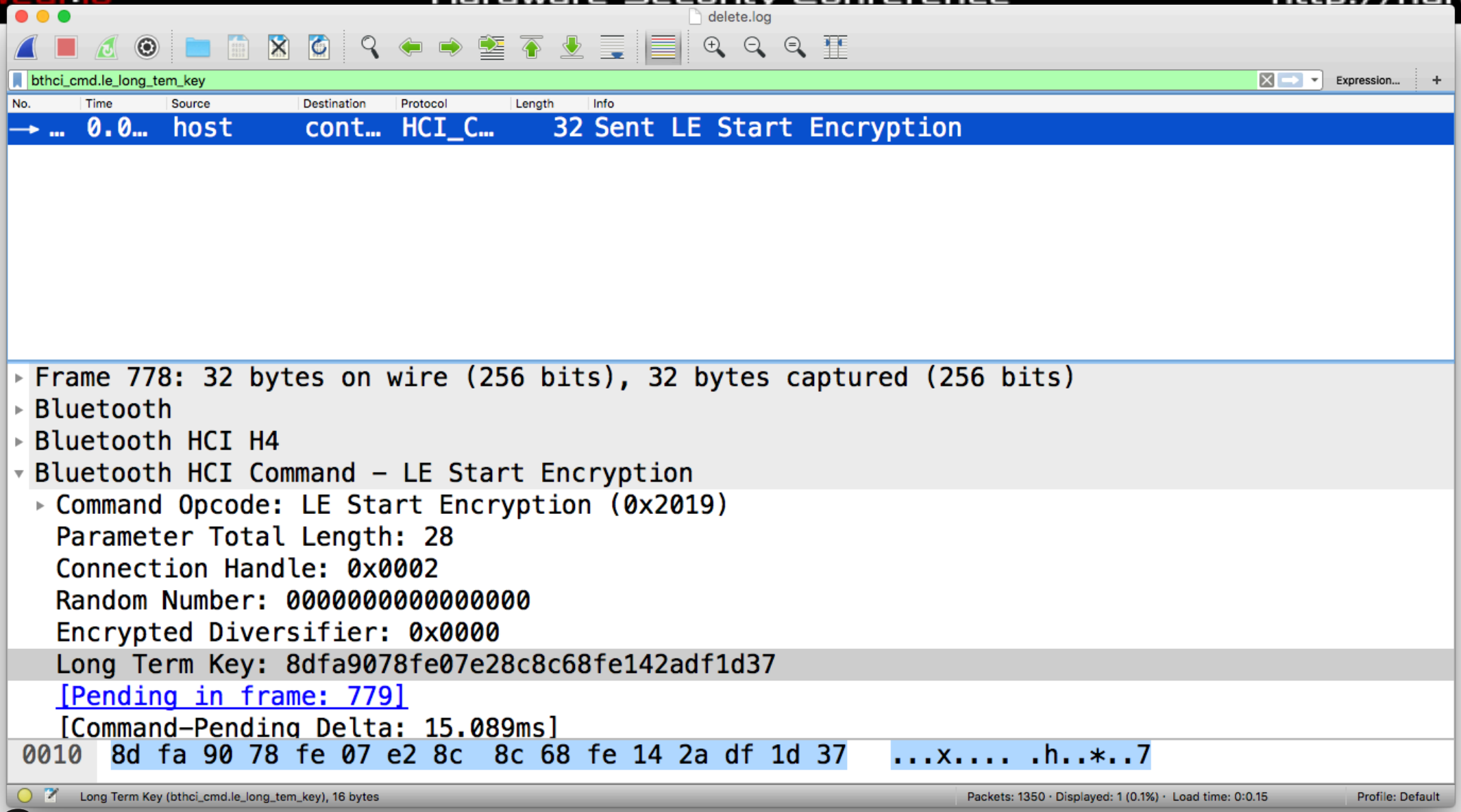


Bluetooth credit card



X-ray by John McMaster (@johndmcmaster)

ICE9 CONSULTING
https://ice9.us/



ICE9 CONSULTING

misc_actions.log

((btatt.opcode.method == 0x12) && (btatt.handle == 0x0018)) || ((btatt.opcode.method == 0x1b) && (btatt.handle == 0x001a))

Expression...

No.	Time	Source	Destination	Protocol	Lengt	Info
3314	316.173622	LgElectr_e7:94:38…	Brillia…	ATT	32	Sent Write Request, Handle: 0x0018 (Unknown: Unknown)
3317	316.259659	Brillian_00:4e:62…	LgElect…	ATT	25	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)
3318	316.306833	LgElectr_e7:94:38…	Brillia…	ATT	32	Sent Write Request, Handle: 0x0018 (Unknown: Unknown)
3321	316.397758	Brillian_00:4e:62…	LgElect…	ATT	26	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)
3322	316.409395	LgElectr_e7:94:38…	Brillia…	ATT	32	Sent Write Request, Handle: 0x0018 (Unknown: Unknown)
3325	316.485582	Brillian_00:4e:62…	LgElect…	ATT	32	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)
3326	316.486765	Brillian_00:4e:62…	LgElect…	ATT	32	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)
3327	316.489184	Brillian_00:4e:62…	LgElect…	ATT	32	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)
3328	316.528282	Brillian_00:4e:62…	LgElect…	ATT	29	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)
3329	316.994640	LgElectr_e7:94:38…	Brillia…	ATT	32	Sent Write Request, Handle: 0x0018 (Unknown: Unknown)
3332	317.068326	Brillian_00:4e:62…	LgElect…	ATT	29	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)
3371	336.015132	LgElectr_e7:94:38…	Brillia…	ATT	32	Sent Write Request, Handle: 0x0018 (Unknown: Unknown)
3374	337.221906	Brillian_00:4e:62…	LgElect…	ATT	25	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)
3407	345.883355	LgElectr_e7:94:38…	Brillia…	ATT	32	Sent Write Request, Handle: 0x0018 (Unknown: Unknown)
3410	346.843388	Brillian_00:4e:62…	LgElect…	ATT	21	Rcvd Handle Value Notification, Handle: 0x001a (Unknown: Unknown)

```
▶ Frame 3575: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)
▶ Bluetooth
▶ Bluetooth HCI H4
▶ Bluetooth HCI ACL Packet
▶ Bluetooth L2CAP Protocol
▼ Bluetooth Attribute Protocol
  ▶ Opcode: Write Request (0x12)
  ▶ Handle: 0x0018 (Unknown: Unknown)
    Value 02b9010131306d69727374206e616d65000000ee
    [Response in Frame: 3577]
```

```
0000  02 0d 00 1b 00 17 00 04  00 12 18 00 02 b9 01 01   ........ ........
0010  31 30 6d 69 72 73 74 20  6e 61 6d 65 00 00 00 ee   10mirst  name....
```

Value (btatt.value), 20 bytes | Packets: 9801 · Displayed: 59 (0.6%) · Load time: 0:0.149 | Profile: Default

```
tx 02be0101303031000000000000000000008f  ....001.............
tx 02bf0101000000000000000000000000000bf  ....................
tx 02ea0101000000000000000000000000000ea  ....................
tx 02c50101303000000000000000000000000c5  ....00..............
tx 02ea0101000000000000000000000000000ea  ....................
tx 02b90101313046697273742061e616d6500000ee  ....10First name....
tx 02b30201313031303134666f6f6f31303b3939388f  ....10104food10;998.
tx 02b3020238373737373f303033303000000000087  ....87777?00300.....
tx 02ea0101000000000000000000000000000ea  ....................
tx 02c50101313000000000000000000000000c4  ....10..............
```

02 XX YY ZZ .. .. WW

XX – opcode

YY – total number of messages

ZZ – current message

WW – checksum

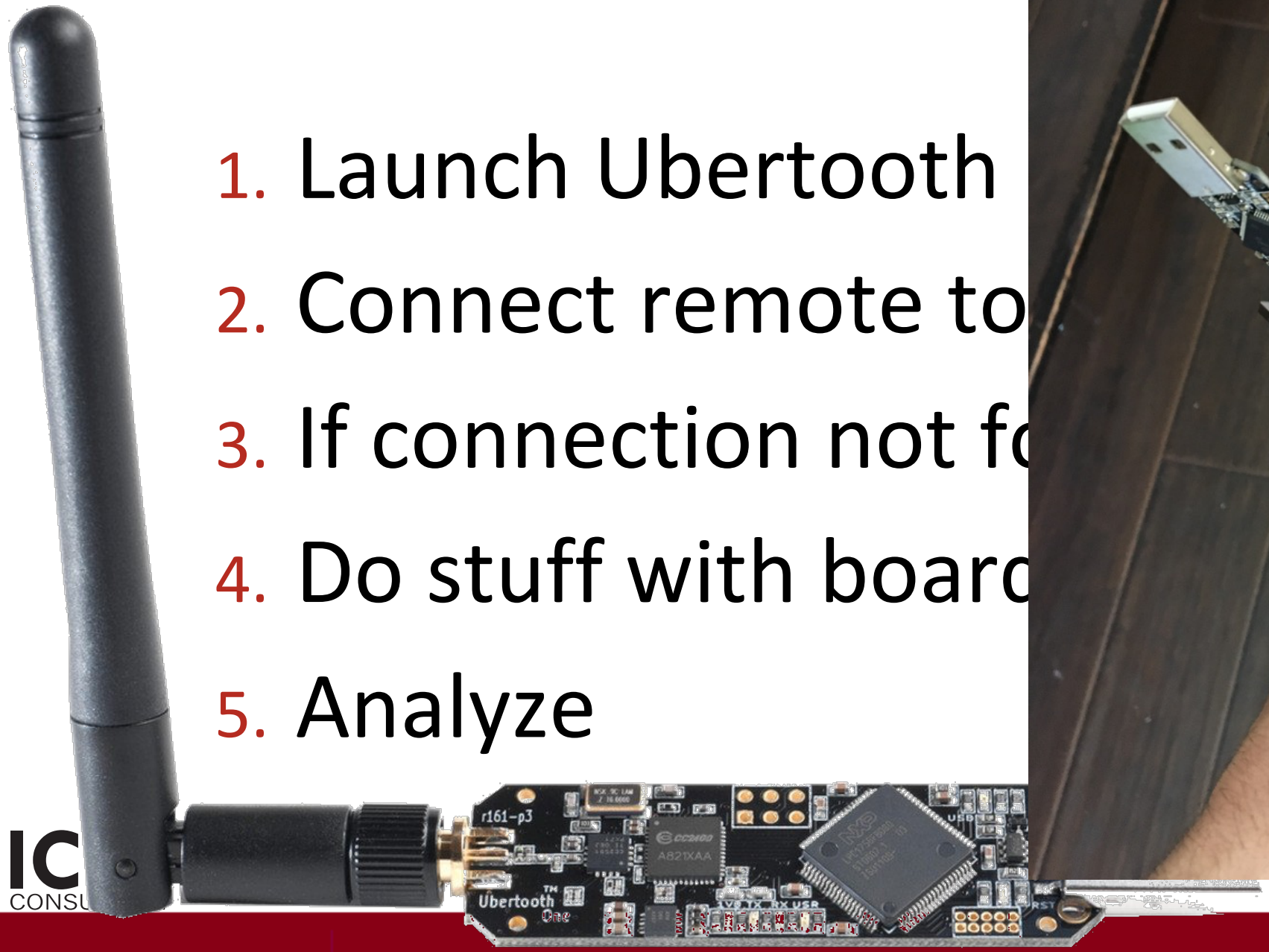
Conclusions: Credit Card

- HCI logging allows us to see encrypted data
- Encryption isn't a silver bullet

Case Study 6: BLE Electric Skateboard

No App!

1. Launch Ubertooth
2. Connect remote to
3. If connection not fo
4. Do stuff with board
5. Analyze



01_connection.pcapng

(btatt.opcode == 0x1d) || (btatt.handle == 0x0013)

No.	Time	Source	Destination	Protocol	Lengt	Info
1830	0.015483	Unknown_0x6477b343	Unknown_0…	ATT	48	UnknownDirection Write Request, Handle: 0x001a (Unknown)
1844	0.067138	Unknown_0x6477b343	Unknown_0…	ATT	48	UnknownDirection Write Request, Handle: 0x001a (Unknown)
1850	0.029968	Unknown_0x6477b343	Unknown_0…	ATT	48	UnknownDirection Write Request, Handle: 0x001a (Unknown)
1856	0.030261	Unknown_0x6477b343	Unknown_0…	ATT	48	UnknownDirection Write Request, Handle: 0x001a (Unknown)
1862	0.029887	Unknown_0x6477b343	Unknown_0…	ATT	48	UnknownDirection Write Request, Handle: 0x001a (Unknown)
1870	0.030028	Unknown_0x6477b343	Unknown_0…	ATT	48	UnknownDirection Write Request, Handle: 0x001a (Unknown)
1878	0.029732	Unknown_0x6477b343	Unknown_0…	ATT	48	UnknownDirection Write Request, Handle: 0x001a (Unknown)
1884	0.030005	Unknown_0x6477b343	Unknown_0…	ATT	48	UnknownDirection Write Request, Handle: 0x001a (Unknown)

No Encryption

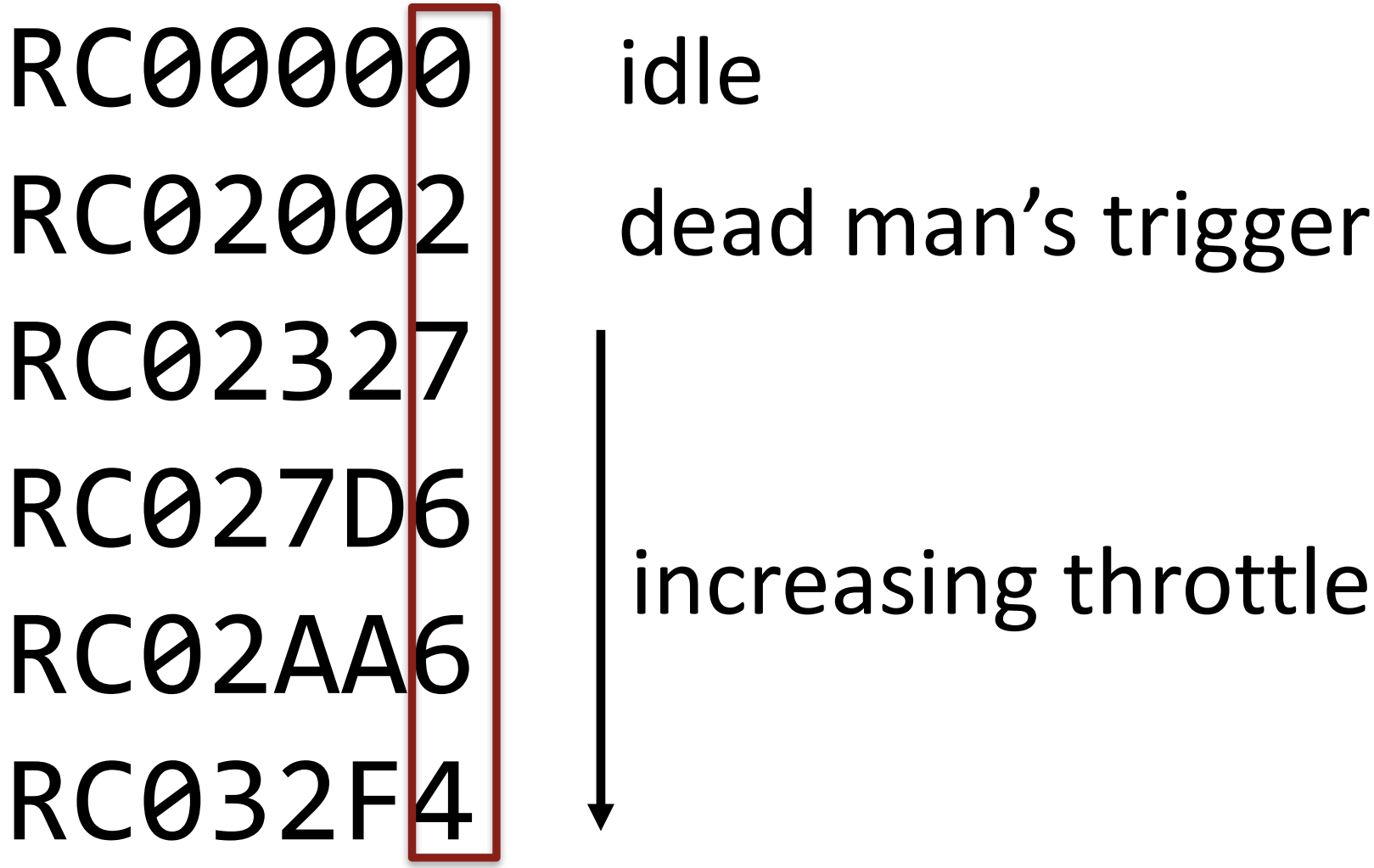
Bluetooth Attribute Protocol
- Opcode: Write Request (0x12)
- Handle: 0x001a (Unknown)
- Value: 524330323332370d
- [Response in Frame: 1874]

```
0000  00 00 18 00 93 00 00 00  36 75 0c 00 00 7e 09 00   ........ 6u...~..
0010  22 b9 4b 7b 80 7f 00 00  43 b3 77 64 0e 0f 0b 00   ".K{.... C.wd....
0020  04 00 12 1a 00 52 43 30  32 33 32 37 0d bd a8 be   ..... RC0 2327....
```

Opcode (btatt.opcode), 1 byte

Packets: 2821 · Displayed:

ICE9 CONSULTING



ICE9 CONSULTING

Conclusions: Skateboard

- Ubertooth is much harder to use than HCI logging
- If using encryption, have to crack

Parting Thoughts

Most Common Security Problems

- No encryption
- Problems with home made encryption
- Debug interfaces left behind
- Incomplete threat modeling

Conclusions

Affordable

RE Process and Tools

Case Studies

Call to Action

Go forth and hack some Bluetooth

Bluetooth Hacking: Tools and Techniques

https://ice9.us/

mike@ice9.us

ICE9
CONSULTING