



AppSec Labs Ltd.
info@appsec-labs.com
<https://appsec-labs.com>

Hacking Bluetooth Low Energy Based Applications



Tal Melamed
Application Security Expert
Tal@appsec.it

The Twelfth International Conference on Internet
Monitoring and Protection
ICIMP 2017
June 25 - 29, 2017- Venice, Italy

- 📄 Key aspects in Bluetooth Low Energy (BLE)
- 📄 How is it different than Bluetooth Classic?
- 📄 Where is the risk?
- 📄 Bluetooth Low Energy Architecture
- 📄 The Security Manager
- 📄 Bluetooth Pairing
- 📄 Generic Attribute Profile (GATT)
- 📄 Man-in-the-Middle (MitM)
- 📄 Related work
- 📄 Possible Mitigations
- 📄 Bibliography

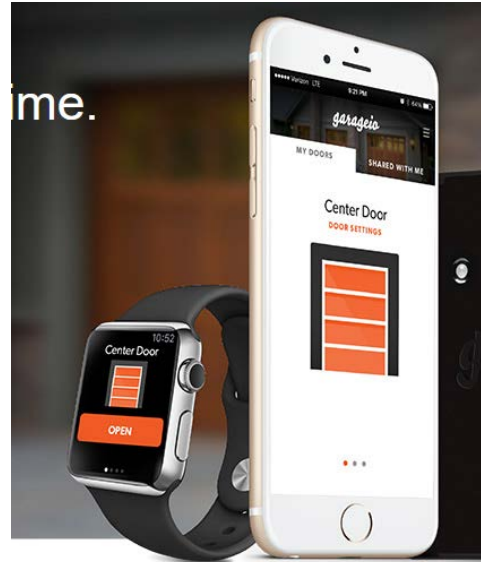
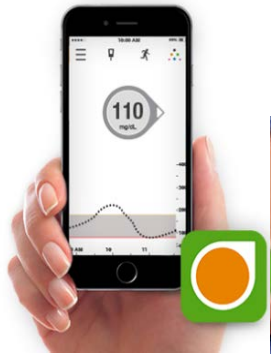
What is Bluetooth Low Energy

- ❏ Bluetooth Low Energy (BLE)
 - ❏ a.k.a Bluetooth Smart, part of Bluetooth 4
- ❏ Designed to be power-efficient
- ❏ Significantly smaller and cheaper.
- ❏ Low cost and ease of implementation lead BLE to be widely used among IoT devices and applications
- ❏ Wearables, sensors, lightbulbs, medical devices, and many other smart-products.
- ❏ 48 billion IoT devices expected by 2021, and Bluetooth—predicted to be in nearly one-third of those devices

Where is the difference?

- ❏ BLE vs BT Classic
 - ❏ Different architecture (Master-Slave)
 - ❏ Different modulation parameters
 - ❏ Different channels
 - ❏ Different channel-hopping scheme
 - ❏ Different packet format
 - ❏ Different packet whitening

Where is the risk?



BLE products can be found in our day-to-day life...



Apps

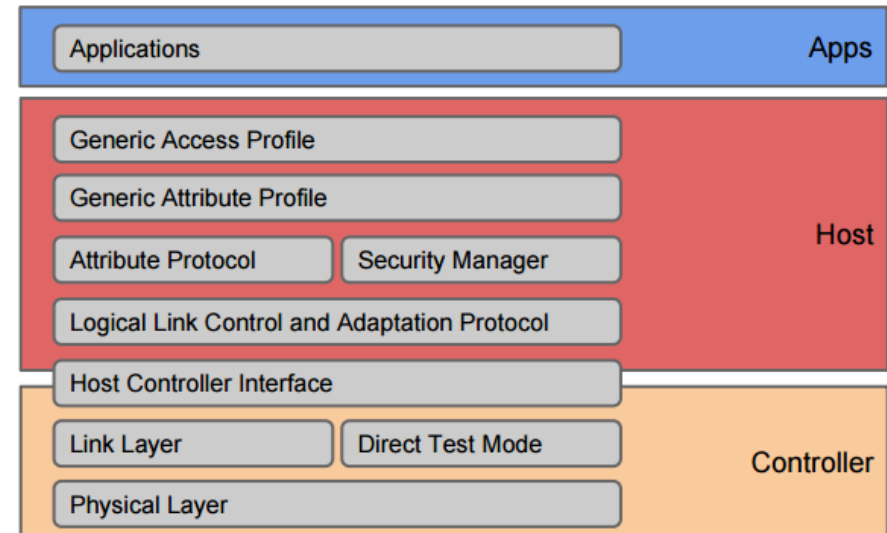
- ▣ Applications are built on top
- ▣ Interacts with host layer only
- ▣ Different API's depending on the application environment

Host

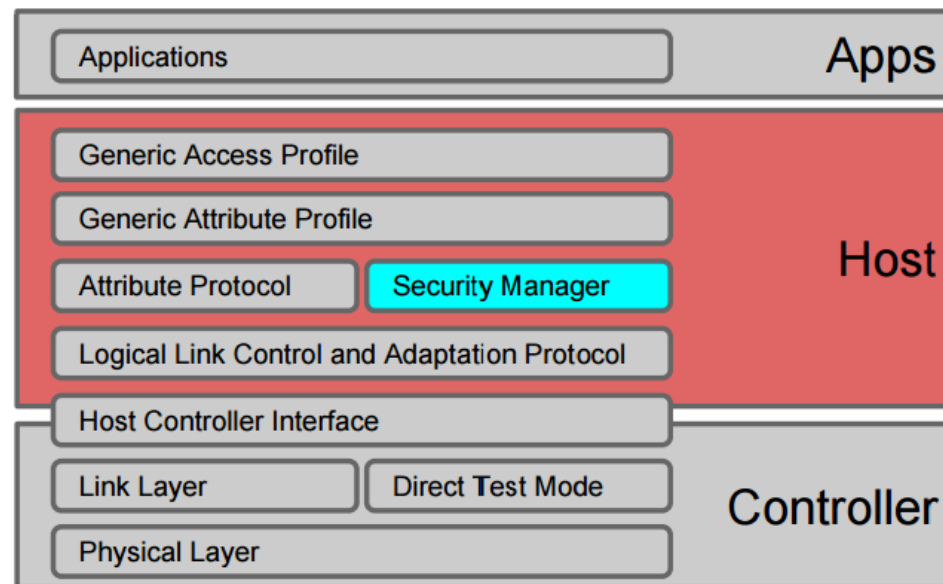
- ▣ Sits on top of the Radio
- ▣ Provides API to application

Controller

- ▣ Radio Control
- ▣ Connection Linking
- ▣ Radio Testing
- ▣ Interface to Host

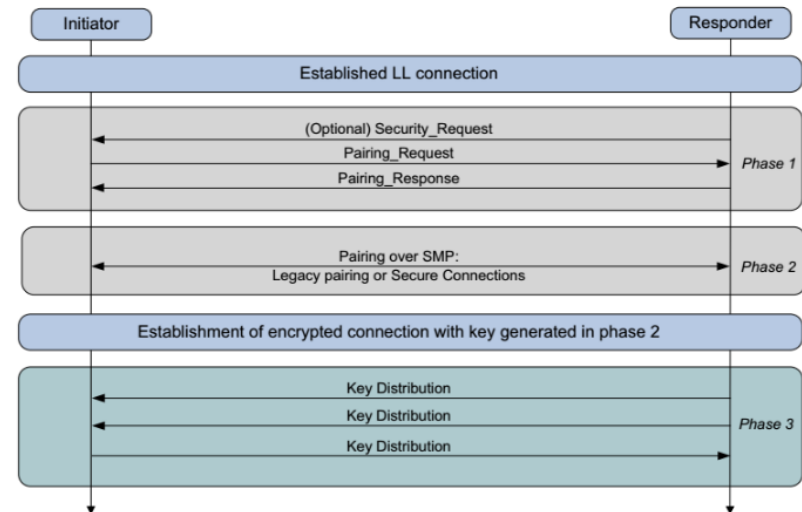


- ☐ Three phase process on connection
 - ☐ Pairing feature exchange
 - ☐ Short term key generation
 - ☐ Transport specific key distribution
- ☐ Implements a number of cryptographic functions



- ❏ Has AES-128 capabilities
- ❏ Uses Key Distribution to share various keys
 - ❏ Bluetooth Smart (4.0) uses an insecure
 - ❏ BLE 4.1/5.0 uses EC-DH for key exchange
- ❏ Pairing encrypts the link using a Temporary Key (TK)
 - ❏ Derived from passkey
 - ❏ Then distribute keys

- Using keys to encrypt the communication
 - The keys can be used to encrypt future reconnections
- Can also verify signed data, or perform random address resolution



3-phase for pairing

- Pairing Feature Exchange
- Short Term Key (STK) Generation (legacy pairing)
 - Long Term Key (LTK) Generation (4.1/5.0 Secure Connections)
- Transport Specific Key Distribution

☐ How to determine the temporary key (TK)?

☐ **JustWorks™**

- ☐ Legacy, most common
- ☐ Devices without display cannot implement other
- ☐ Its actually a key of zero, that's why it just works...

☐ **6-digit PIN**

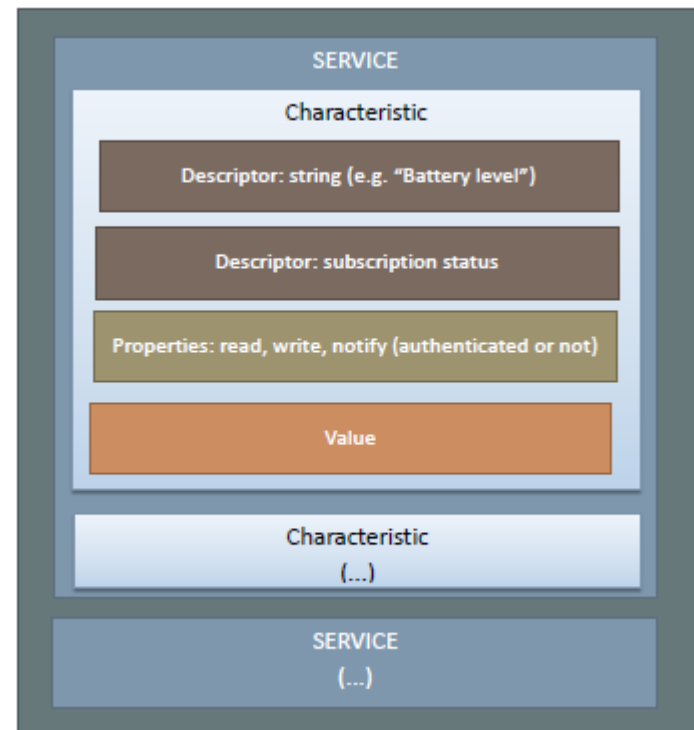
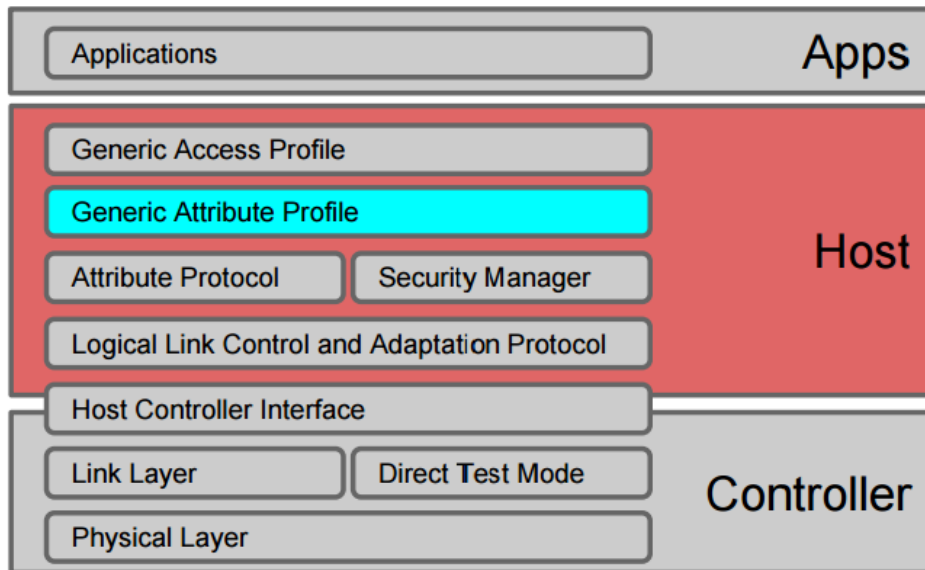
- ☐ In case the device has a display
- ☐ 1 million options (BF-able)

☐ **Out of band (OOB)**

- ☐ Does not share secret key over the 2.4 GHz band (used by protocol)
- ☐ Makes use of other mediums (e.g. NFC)
- ☐ Once secret keys are exchanged, encrypts the channel
- ☐ Not common, barely used

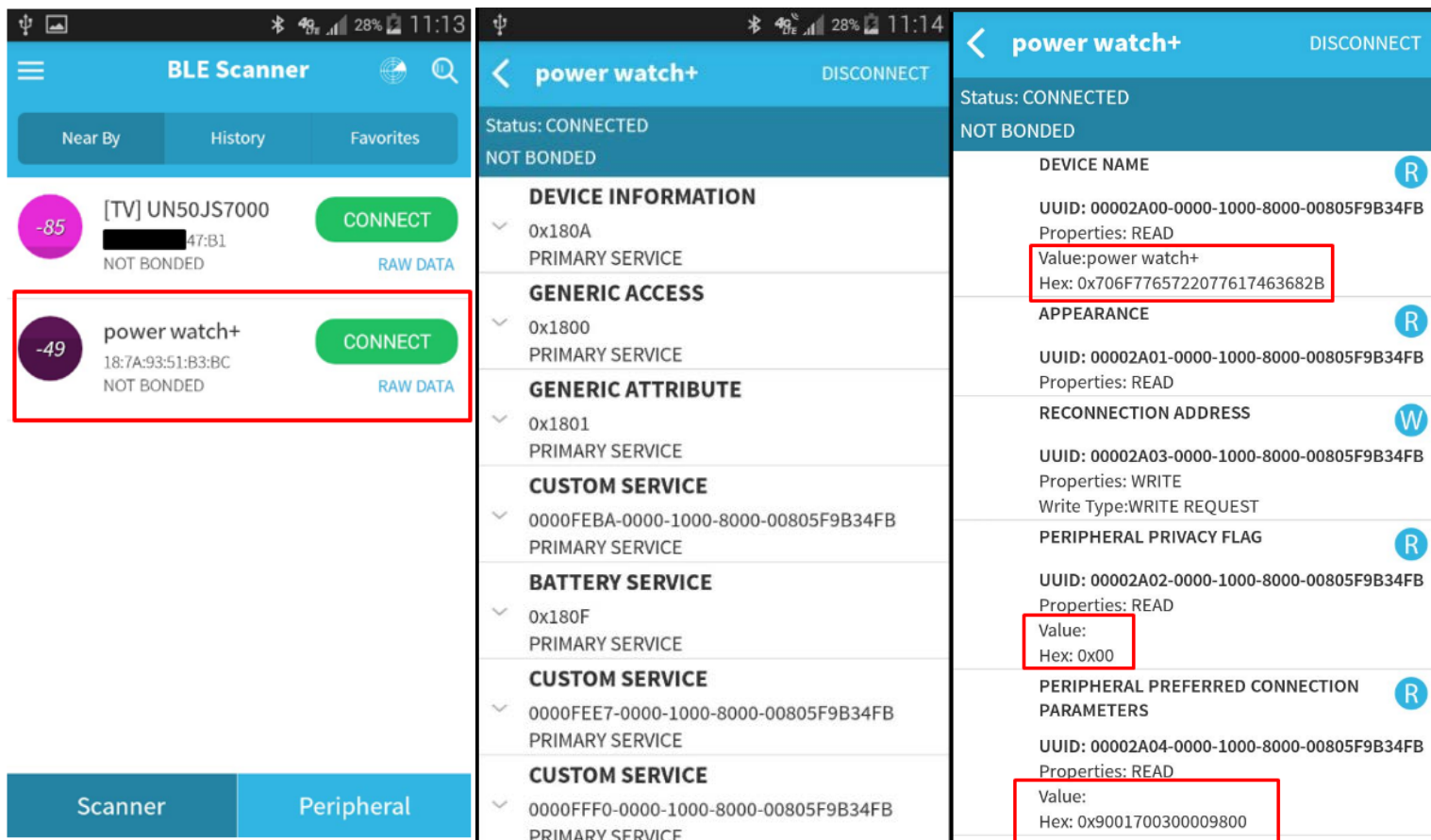
Generic Attribute Profile (GATT)

- Services & characteristic are identified by an associated UUID
- A characteristic contains a single value (“attribute”)
 - Can be read, written to or subscribed for notifications



Discovering Services - Example

Any BLE scanner app, downloaded from the store, can read data from and write data to the smart-device



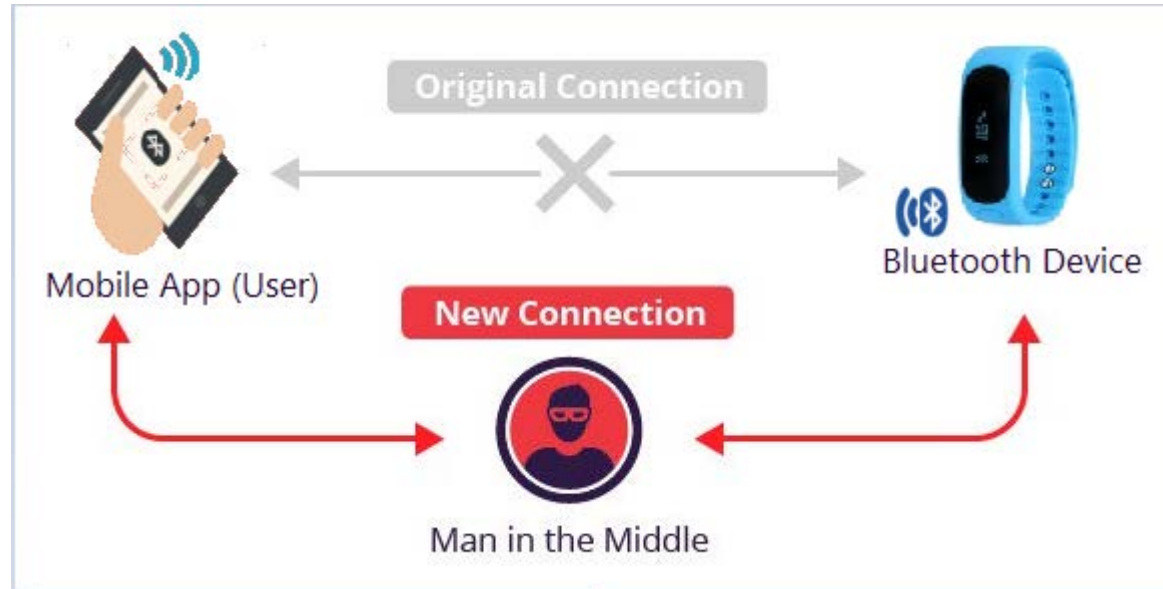
The image shows three screenshots from a mobile application demonstrating BLE service discovery and connection details for a device named 'power watch+'.

Left Screenshot: BLE Scanner
 Shows a list of nearby BLE devices. The 'power watch+' device is highlighted with a red box. It has a signal strength of -49 dBm and MAC address 18:7A:93:51:B3:BC. A 'CONNECT' button is visible next to it.

Middle Screenshot: Device Information
 Shows the details for the 'power watch+' device. It is currently 'CONNECTED' but 'NOT BONDED'. The services listed are:
 - **0x180A PRIMARY SERVICE**
 - **0x1800 PRIMARY SERVICE**
 - **0x1801 PRIMARY SERVICE**
 - **0000FEBA-0000-1000-8000-00805F9B34FB PRIMARY SERVICE**
 - **0x180F PRIMARY SERVICE**
 - **0000FEE7-0000-1000-8000-00805F9B34FB PRIMARY SERVICE**
 - **0000FFF0-0000-1000-8000-00805F9B34FB PRIMARY SERVICE**

Right Screenshot: Service Details
 Shows the details for the 'power watch+' service. It is 'CONNECTED' but 'NOT BONDED'. The details include:
 - **DEVICE NAME**: power watch+ (Value: power watch+, Hex: 0x706F7765722077617463682B)
 - **APPEARANCE**: (Value: (empty), Hex: 0x00)
 - **PERIPHERAL PRIVACY FLAG**: (Value: (empty), Hex: 0x00)
 - **PERIPHERAL PREFERRED CONNECTION PARAMETERS**: (Value: (empty), Hex: 0x9001700300009800)

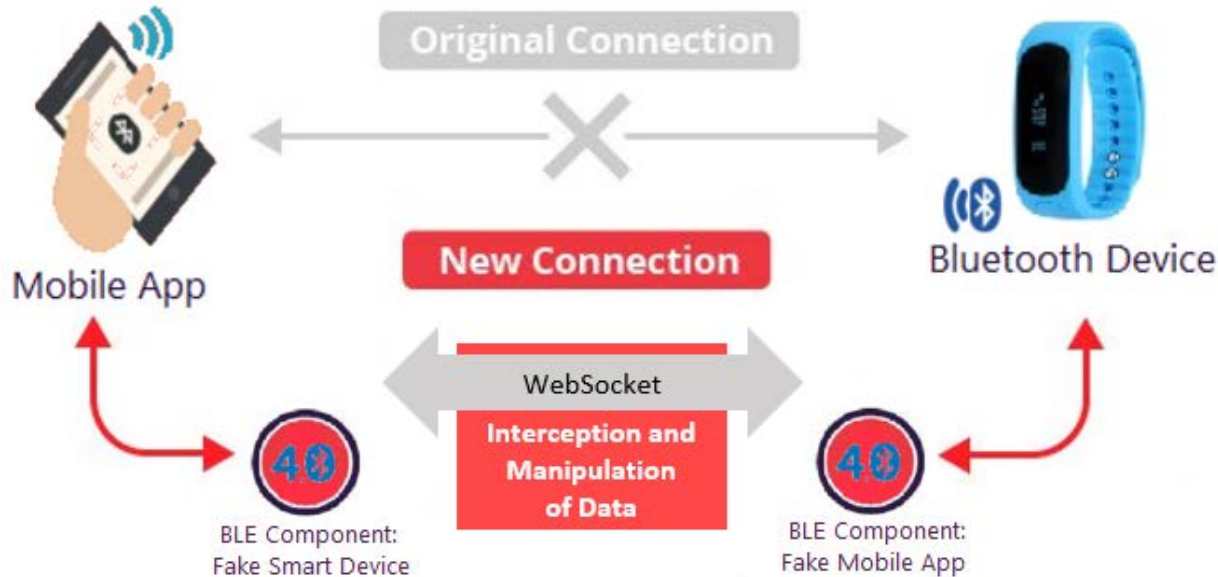
Normal Man-in-the-Middle (MitM)



Why normal MitM won't work?

- ▣ A BLE adapter cannot serve as both ends
- ▣ One will have to serve as the client (app)
- ▣ Another as the server (ble device)

BLE Man-in-the-Middle (MitM)



- ☐ After each BLE-adapter (component) is connected to the designated device – they communicate with each other over WebSocket
- ☐ Which gives them the ability to serve as MitM

What to we need for MitM

- CSR 4.0 dongle x2
- Works as Slave/Master



- Download Kali-linux VM and Clone

The image shows two VMware Workstation windows. The left window, titled 'Kali-Linux-2011-SLAVE', has the 'Removable Devices' menu open, showing a list of devices including 'Cambridge Silicon Radio CSR8510 A10', which is highlighted with a red box. The right window, titled 'Kali-Linux-2016.1-vm-i686', shows a terminal window with system logs. A red box highlights the line 'usb 3-1.2: Product: CSR8510 A10' in the logs. Below the logs, the terminal shows the command 'usbcore: registered new interface driver btusb'.

Hooking events using GATTacker

📱 Hooking into smart-watch sports counter and modifying the data (kilometrage) sent from the smart watch into the device

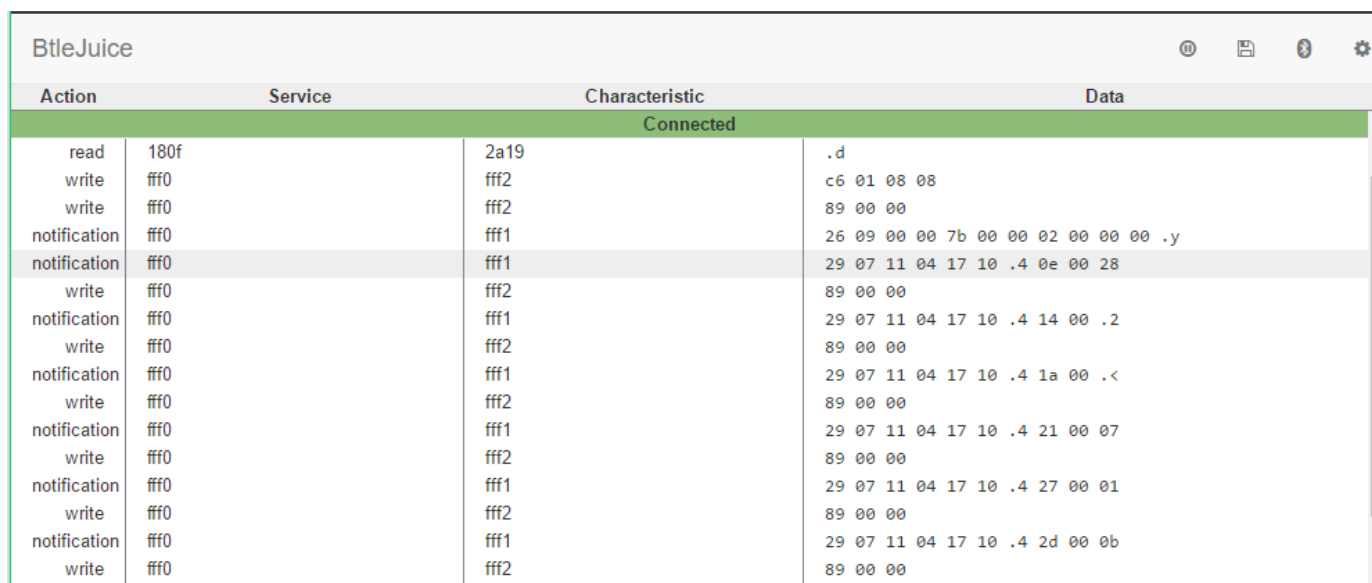
The image displays a mobile application interface on the right and a terminal window on the left. The mobile app shows a blue screen with a white running icon, the text "8,021.611 KM", and a "FINISH" button. The terminal window shows the GATTacker configuration in a JSON file and the hook function implementation in a JavaScript file.

```
i07u0000i70ii.srv.json + (/usr/...modules/gattacker/devices) - VIM
{
  "uuid": "fff3",
  "name": null,
  "properties": [
    "notify"
  ],
  "value": "",
  "descriptors": [
    {
      "handle": 68,
      "uuid": "2902",
      "value": ""
    }
  ],
  "startHandle": 66,
  "valueHandle": 67,
  "hooks": {
    "dynamicNotify": "TreadMill_Hook"
  }
}

pos.js (/usr/local/lib/node_modules/gattacker/hookFunctions) - VIM
function TreadMill_Hook(peripheralId, service, characteristic, type, data, notifyEmitter, callback){
  console.log('  dynamic notify hook'.yellow);
  datastr = data.toString('hex');
  // check if data is related to ThreadMill Exercise...
  if (datastr.startsWith("970908")){
    console.log('  original data was:' + datastr);
    // replace characters in position 15-16 from "00" to "11"
    datastr = datastr.replace("9709080000000000", "9709080000000011");
    console.log('  new data is:' + datastr);
  }
  callback(null, new Buffer(datastr, 'hex'));
}
```

Bluetooth Smart (LE) Man-in-the-Middle framework

<https://github.com/DigitalSecurity/BtleJuice>



Action	Service	Characteristic	Data
Connected			
read	180f	2a19	.d
write	fff0	fff2	c6 01 08 08
write	fff0	fff2	89 00 00
notification	fff0	fff1	26 09 00 00 7b 00 00 02 00 00 00 .y
notification	fff0	fff1	29 07 11 04 17 10 .4 0e 00 28
write	fff0	fff2	89 00 00
notification	fff0	fff1	29 07 11 04 17 10 .4 14 00 .2
write	fff0	fff2	89 00 00
notification	fff0	fff1	29 07 11 04 17 10 .4 1a 00 .<
write	fff0	fff2	89 00 00
notification	fff0	fff1	29 07 11 04 17 10 .4 21 00 07
write	fff0	fff2	89 00 00
notification	fff0	fff1	29 07 11 04 17 10 .4 27 00 01
write	fff0	fff2	89 00 00
notification	fff0	fff1	29 07 11 04 17 10 .4 2d 00 0b
write	fff0	fff2	89 00 00

Replay & on-the-fly data modification

Web interface

Replay Attack using BtleJuice

Remote control over the victim's mobile using *Replay Attack*

Taking pictures

Playing music

The screenshot shows the BtleJuice application interface. At the top, it says "BtleJuice". Below that is a table with columns: Action, Service, Characteristic, and Data. The table is currently empty. In the center, there is a modal window titled "Replay notification" with a pencil icon. It contains the following fields: Service: fff0, Characteristic: fff1, and Data: f6 01 01. There are "Notify" and "Close" buttons at the bottom of the modal. On the right side of the interface, there is a list of Bluetooth services and characteristics with their UUIDs. A context menu is open over the entry "29 07", showing "Replay" and "Set hook" options. The "Replay" option is highlighted.

Action	Service	Characteristic	Data
read	180f	2a19	
notification	fff0		
notification	fff0		
write	fff0		
notification	fff0		
write	fff0		
notification	fff0		

Replay notification

Service: fff0

Characteristic: fff1

Data: f6 01 01

Notify Close

.d

f6 01 01

f6 01 01

89 00

29 07

9b 04

2b 00 00

Replay

Set hook

☐ Attacks on advertisements

- ☐ The attacker clones the advertisement and broadcasts the fake device
- ☐ The device will try to connect and fail

☐ **Countermeasures:**

- ☐ Do not rely on received packets for critical functionality

☐ Attacks on exposed services

- ☐ If the device offers services possible to access without authentication, an attacker can:

- ☐ Brute-force data (e.g. guessing the password)
- ☐ Fuzzing (Sending improper values to characteristics)
- ☐ Logic vulnerabilities

☐ **Countermeasures:**

- ☐ Restrict access to services (e.g. least privilege)
- ☐ Perform input validation
- ☐ Time-limited provisioning (expose services only for a limited time after power-up, or dedicated button)

Attacks on Pairing

- ▣ An attacker can trick the user into re-initiation of the pairing using Jamming, cloning, etc.

Countermeasures:

- ▣ “Something you have” (e.g. allow pairing initiation only after performing the required action on the smart device - e.g. push a dedicated button)
- ▣ Mobile app should warn when wrong MAC is used.

Man-in-the-Middle (MitM) attack

- ▣ Unencrypted transmission can be intercepted via passive eavesdropper
 - ▣ Exposing sensitive data (health data, passwords, etc.)
 - ▣ Data can be tampered with
 - ▣ Replay attack (e.g. unlock device)

Countermeasures:

- ▣ Encrypt data in transit, sign it and validate the input

Summary

- ☐ This poster confirms that BLE is insecure and vulnerable against passive eavesdropping.
- ☐ In particular, I have shown that a passive eavesdropping can easily become an active MitM attack that enables a possible hacker not only to listen to the communication, but also to intercept and manipulate the data.
- ☐ By performing a MitM attack, hackers can even control from remote the mobile device used to communicate with the Bluetooth smart device.
- ☐ With the release of the Bluetooth Core Specification version 4.2, BLE Security has been significantly improved by the new LE Secure Connections pairing model
- ☐ Additional security and privacy related features are added in the Bluetooth Core Specification v5, recently released by Bluetooth.
- ☐ It is vital to be aware and fully understand the limitations of the smart devices that we use rather than blindly relying on them.
- ☐ It is essential to implement security protections on the application-side to protect against malicious activity, by implementing additional security controls, such as data encryption, strong authentication and authorization mechanisms, and other security best practices.

- ❏ Bluetooth, S.I.G, SIG introduces Bluetooth Low Energy wireless technology, the next generation of Bluetooth wireless technology, *press release*, 2009.
- ❏ Ryan, Mike. Bluetooth: With Low Energy Comes Low Security, *WOOT*, 2013. Online, <https://www.usenix.org/system/files/conference/woot13/woot13-ryan.pdf>. Accessed on: 30 Apr. 2017.
- ❏ Bluetooth S.I.G., Proprietary Information Security, Bluetooth Low Energy. Online, <https://www.bluetooth.com/~media/files/specification/bluetooth-low-energy-security.ashx>. Accessed on: 30 Apr. 2017.
- ❏ Bluetooth Specifications. Online, <https://www.bluetooth.com/specifications>. Accessed on: 30 Apr. 2017.
- ❏ Jasek S., GATTacking Bluetooth Smart devices, BlackHat USA 2016. Online, <http://gattack.io/whitepaper.pdf>. Access on: 30 Apr 2017.
- ❏ Cauquil D., BtleJuice: the Bluetooth Smart MitM Framework, DEF CON 24 Internet of Things Village, 2016. Video Online, <https://www.youtube.com/watch?v=lcn07TclnS0>. Accessed on: 30 Apr. 2017.
- ❏ Melamed T., R U aBLE? BLE Application Hacking, *OWASP*, 2017. Online, <http://sl.owasp.org/melamed17>. Accessed on: 30 Apr. 2017.