R U aBLE?
BLE Application Hacking

Tal Melamed
Application Security Expert
Tal@AppSec-Labs.com
- What is BLE?
- BLE security
- BLE pairing
- Discovering services
- Cloning the device
- BLE MitM
- Available tools
- What’s next?
about://me

- Tech Lead @ AppSec Labs
  Tal@AppSec-Labs.com
- Application Security Expert
- Follow me @ appsec.it
- https://github.com/nu11p0inter
- http://sl.owasp.org/talmelamed
AppSec Labs

WHO ARE WE?

WHAT DO WE WANT?

https://appsec-labs.com/

IoT

Secured apps!

OWASP Israel
AppSec Labs provides its high end services to the following industry vectors:

- High Tech software development
- Banking and financing
- National security
- IoT
- Cloud
- Pharmaceuticals
- Commerce
- Travel and transport
- IT Security products
- Biometrics
- Education
- Gaming
- Government
- Telecommunications
Before we start

- AppSec Labs – Securing the IoT
  - Sounds like a buzz
  - PenTesting RF Operated Devices, OWASP 2016
  - Azure IoT partner
  - https://appsec-labs.com/iot-security/

- What we offer?
  - SDLC, Training, Penetration-testing, Consulting

- This presentation is a result of a preliminary research on BLE security and part of a wide research for IoT Security
What is BLE?

- Bluetooth Low Energy
  - a.k.a Bluetooth Smart, part of Bluetooth 4
  - Designed to be power-efficient
  - Claimed to operate “month to years” on a single battery
  - Significantly smaller and cheaper.
  - Operates at 2.4 GHz (same as Bluetooth Classic)
  - Short range (<100m)

- Low cost and ease of implementation lead BLE to be widely used among IoT devices and applications

- Used among medical, industrial and government equipment
  - 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
To save you some questions

- BLE vs BT Classic
  - Different architecture (Master-Slave)
  - Different modulation parameters
  - Different channels
  - Different channel-hopping scheme
  - Different packet format
  - Different packet whitening
J&J warns diabetic patients: Insulin pump vulnerable to hacking
And for me?

- OMG, your baby has 41° - take him to the hospital!

- Your blood sugar is way up. You should get your insulin...
Ok, medical but...
What else?

- Opens the Garage door
- Are you drunk?
- Possibly anything!
BLE Architecture

- **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 applications

- **Controller**
  - Radio Control
  - Connection Linking
  - Radio Testing
  - Interface to Host
BLE Security - Security Manager

- Three phase process on connection
  - Pairing feature exchange
  - Short term key generation
  - Transport specific key distribution
- Implements a number of cryptographic functions
- Memory and processing requirements are lower for responding (saves power)
BLE Security

- Uses AES-128 with CCM encryption engine

- Uses Key Distribution to share various keys
  - Identity Resolving Key is used for privacy
  - Signing Resolving Key provides fast authentication without encryption
  - Long Term Key is used

- Pairing encrypts the link using a Temporary Key (TK)
  - Derived from passkey
  - Then distribute keys

- Asymmetric key model
  - Slave gives keys to master with a diversifier
  - Slave can then recover keys from the diversifier
BLE Security: Pairing

- Performed to establish keys in order to encrypt a link.
- The keys can be used to encrypt a link in future reconnections, verify signed data, or perform random address resolution.

3-phase for paring
- Pairing Feature Exchange
- Short Term Key (STK) Generation (legacy pairing)
- Long Term Key (LTK) Generation (Secure Connections)
- Transport Specific Key Distribution
BLE Security: Pairing

How to determine the temporary key (TK)?

- **Just Works**
  - 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
  - 1m options (BFable)

- **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 (understatement – haven’t seen one yet)

“None of the pairing methods provide protection against a passive eavesdropper” - Bluetooth Core Spec
"A future version of this specification will include elliptic curve cryptography and Diffie-Hellman public key exchanges..."

☑ Well, that’s too bad, I already bought the device!
So? Is it secure or not?

- In practice, ~80% of tested devices do not implement BLE-layer encryption
- Mobile apps cannot control the pairing (OS level)

Why?

- As always, security is left behind (cost, time, etc.)
- Multiple users/apps using the same devices
- Access sharing
- Backups to the cloud
- Public access devices (e.g. cash register)
- Hardware, software or even UX compatibilities/requirements
GATT

- Generic Attribute Profile (GATT) used by BLE to communicate with each other

- Client Server Architecture
  - Built on top of ATT (Attribute Protocol)
  - GATT Server stores data using ATT
  - GATT Server accepts ATT requests to serve and save attributes

- Organized in data objects
  - Profiles
  - Services
  - Characteristics
Services & characteristic are identified by an associated UUID

A characteristic contains a single value ("attribute")
- Can be read, written to or subscribed for notifications
Typical flow

1. Start scanning for advertisements
2. Specific advertisement received
3. Advertise
4. Connect the advertising device (MAC)
5. Further communication
How does that work?

Hello everyone!
I am a BLE device
and these are my services
BLE cmd basics

- # hciconfig hci0 up/down/reset
- # hcitool lescan
- # hcidump -x -t hci0
root@AppSecLabs:~# gatttool --device=18:7A:93:51:B3:BC --interactive
Attempting to connect to 18:7A:93:51:B3:BC
Connection successful
handle: 0x0002, char properties: 0x02, char value handle: 0x0003, uuid: 00002a00-0000-1000-8000-0080f9b34fb
handle: 0x0004, char properties: 0x02, char value handle: 0x0005, uuid: 00002a01-0000-1000-8000-0080f9b34fb
handle: 0x0006, char properties: 0x08, char value handle: 0x0007, uuid: 00002a03-0000-1000-8000-0080f9b34fb
handle: 0x0008, char properties: 0x02, char value handle: 0x0009, uuid: 00002a02-0000-1000-8000-0080f9b34fb
handle: 0x000a, char properties: 0x02, char value handle: 0x000b, uuid: 00002a04-0000-1000-8000-0080f9b34fb
handle: 0x000d, char properties: 0x02, char value handle: 0x000e, uuid: 00002a05-0000-1000-8000-0080f9b34fb
handle: 0x000f, char properties: 0x02, char value handle: 0x0012, uuid: 00002a25-0000-1000-8000-0080f9b34fb
handle: 0x0014, char properties: 0x02, char value handle: 0x0015, uuid: 00002a29-0000-1000-8000-0080f9b34fb
handle: 0x0017, char properties: 0x02, char value handle: 0x0018, uuid: 00002a23-0000-1000-8000-0080f9b34fb

[18:7A:93:51:B3:BC][LE]> char-read-uuid 0000fff5-0000-1000-8000-0080f9b34fb
handle: 0x0048 value: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12
Characteristic value/descriptor: 02 48 00 f5 ff
Error: Characteristic Write Request failed: Attribute value length is invalid
Characteristic value was written successfully
Cloning the device

- Scan devices
- Listen to advertisements
- Clone
- Advertise cloned services
- Let client connect to your “fake device”.
- MitM!

If you kill your clone

is it murder or suicide?
Normal MitM

1. Needs to advertise more frequently
2. Needs to keep the connection to the original device
3. While connected - does not advertise

Start scanning for advertisements
Specific advertisement received
Connect the advertising device (MAC)
Further communication
What do we need?

- CSR 4.0 dongle x2
  - One will service as the client
  - The other as the server
  - $2.70 in aliexpress
Desired MitM for BLE
Tampering
Tampering with data using Burp

<table>
<thead>
<tr>
<th>#</th>
<th>URL</th>
<th>Direction</th>
<th>Edited</th>
<th>Length</th>
<th>SSL</th>
<th>Time</th>
<th>Listen...</th>
</tr>
</thead>
<tbody>
<tr>
<td>772</td>
<td><a href="http://10.0.0.107:2846/">http://10.0.0.107:2846/</a></td>
<td>Incoming</td>
<td>✓</td>
<td>150</td>
<td></td>
<td>20:21:00 15 Jan 2017</td>
<td>2846</td>
</tr>
<tr>
<td>482</td>
<td><a href="http://10.0.0.107:2846/">http://10.0.0.107:2846/</a></td>
<td>Incoming</td>
<td>✓</td>
<td>150</td>
<td></td>
<td>20:00:50 15 Jan 2017</td>
<td>2846</td>
</tr>
</tbody>
</table>

Original message:
```
{"type": "read", "peripheralId": "187a9351b3bc", "serviceUuid": "fff0", "characteristicUuid": "fff3", "data": "9709040000000000000000000000000000001c18", isNotification":true}
```

Edited message:
```
{"type": "read", "peripheralId": "187a9351b3bc", "serviceUuid": "fff0", "characteristicUuid": "fff3", "data": "9709040000000000000000000000000000001c18", isNotification":true}
```
gattacker

BLE (Bluetooth Low Energy) security assessment using Man-in-the-Middle

https://github.com/securing/gattacker
BtleJuice

- Bluetooth Smart (LE) Man-in-the-Middle framework
  https://github.com/DigitalSecurity/BtleJuice
- Web interface

Replay & on-the-fly data modification
Thanks

- Sławomir Jasek (gattacker, BH USA 2016)
- Damien Cauquil (Btlejuice, Hack.lu 2016)
- Zero_Chaos & Granolocks (DEF CON 24)
- Mike Ryan (lacklustre.net)
- Sandeep Mistry (github.com/sandeepmistry)
What’s next?

Bluetooth 5

- Significantly increasing the range, speed and broadcast messaging capacity of Bluetooth applications
  - 4x the range, 2x the speed and 8x the broadcasting message capacity
  - “connectionless” IoT, advancing beacon and location-based capabilities
  - Improved interoperability and coexistence with other wireless technologies
  - Advance IoT experience by enabling simple and effortless interactions across the vast range of connected devices

https://www.bluetooth.com/bluetooth5
Stay tuned: https://appsec-labs.com/iot-security/