Hacking challenge: steal a car!
Your "local partner in crime"

Sławomir Jasek
• IT security expert
• since 2005, and still loves this job

Agenda
• BLE vs security
• How to hack the car
• New tool
• Vulnerabilities examples
  • Smart lock
  • Anti-theft device
  • Mobile PoS
  • Other gadgets
• MITM encrypted BLE?
• What can we do better
Bluetooth Smart? (aka Low Energy, 4...)

• Probably most thriving IoT technology
  • Wearables, sensors, home automation, household goods, medical devices, door locks, alarms, banking tokens, smart every-things...

• Completely different than previous Bluetooth
BLE (v4.0) security: encryption

• Pairing (once, in a secure environment)
  • JustWorks (R) – most common, devices without display cannot implement other
  • 6-digit PIN – if the device has a display
  • Out of band – not yet spotted in the wild

• "Just Works and Passkey Entry do not provide any passive eavesdropping protection"

• Establish Long Term Key, and store it to secure future communication ("bonding")

Mike Ryan, https://www.lacklustre.net/bluetooth/
BLE (v4.0) security in practice

• 8 of 10 tested devices do not implement BLE-layer encryption
• "Forget" to do it, or do not consider clear-text transmission a problem
• The pairing is in OS level, mobile application does not have full control over it
• It is troublesome to manage with requirements for:
  • Multiple users/application instances per device
  • Access sharing
  • Cloud backup
  • Public access devices (e.g. cash register)
• Other hardware/software/UX problems with pairing
BLE (v4.0) security in practice

• Security in "application" layer (GATT)
• Various authentication schemes
  • Static password/key
  • Challenge-response (most common)
  • PKI
• Own crypto, based usually on AES
• No single standard, library, protocol
How Secure is [redacted]?

[redacted] uses a combination of hardware and technology to ensure the device is secure.

- **Bluetooth:** [redacted] uses AES 128-bit encryption, the same encryption used by the military to protect documents with confidential and secret security levels.

  By using industry leading Bluetooth 4.0 that utilizes 128-bit encryption, and our very own PKI technology with cryptographic key exchange protocols, [redacted] is safe from criminals, hackers, and thieves.

To protect your transactions from unauthorised access by third parties, [redacted] operates in accordance with the highest card payment industry security standards:

- **PCI-DSS (Payment Card Industry Data Security Standard)** is the highest card security standard used in the credit card industry concerning data transfer and data storage.

- **SSL (Secure Sockets Layer) and TLS (Transport Layer Security)** are 'encryption protocols' that protect data that is transmitted over the internet. We are using a 256-bit encryption, the highest possible level at present.

- **PGP (Pretty Good Privacy)** is an international standard for secure personal data storage.

After 87 years of home security innovations, millions of families rely on [redacted] for peace of mind. [redacted]'s long-time leadership and advancements in residential door lock security have now been enhanced with secure authentication technology. Resulting in [redacted] engineered for both maximum security and performance.
Regular users convinced

View full report in Google Trends
So, how to attack the BLE car lock?

• Remote relay?

Figure 3. The relay with antennas, cables and an (optional) amplifier.

Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars
So, how to attack the BLE car lock?

• Remote relay?
• Jamming?
• Brute force?

“It’s like they designed the protocol itself to stop us from doing this exact thing”
Richo Healey, Mike Ryan – Hacking Electric Skateboard, Defcon 23
http://greatscottgadgets.com/ubertoothone/
So, how to attack the BLE car lock?

• Remote relay?
• Jamming?
• Brute force?
• BLE sniffing?
• Mobile app analysis?
• ...
• MITM?

http://greatscottgadgets.com/ubertoothone/
Man in the Middle?
How to MITM: isolate the signal?
How to MITM?

**Stronger signal?**
- Class 1 adapter? +8dBm, 100m range

"little difference in range whether the other end of the link is a Class 1 or Class 2 device as the lower powered device tends to set the range limit"

https://en.wikipedia.org/wiki/Bluetooth

**More signals?**

And how to handle them in a single system?
Typical connection flow

1. Start scanning for advertisements
2. Advertise
3. Specific advertisement received, stop scanning
4. Connect the advertising device (MAC)
5. Further communication
Start scanning for advertisements

Specific advertisement received, stop scanning

Connect the advertising device (MAC)

Further communication

Advertise more frequently

MITM?

Keep connection to original device. It does not advertise while connected ;)

MITM
New tool - architecture

- Advertise (high freq)
- Offer exact services
- Forward req/resp
- Data interception and manipulation
- Gather advertisement and services data for cloning
- Keep connection to original device forward req/resp

Get serv services
websockets
Advertise
Get serv services
New BLE MITM Tool – a must have for IoT tester!

- Open source
- Only $10 BT4 USB dongle needed
- Works on Raspberry or any Linux
- Node.js
- Websockets
- Modular design

- And a cool logo!
Car hacking challenge: authentication

AES("LOGIN", AES(Challenge, key))

Get "Challenge"

Random challenge

AES("LOGIN", AES(Challenge, key))

NOT ENCRYPTED: Open, Close...
Authentication: attack?

Get "Challenge"

Random challenge

AES("LOGIN", AES(Challenge, key))

MITM

Other cmd
Other commands (based on mobile app):

- `initConfigMode` – initiate the configuration – overwrite the keys

- `initiateDataTransfer` – dump the whole configuration (including all keys)
PRNG?

- Is there any function which allows to generate a random number?

- There is no function to do this. However, there is a reasonably good alternative (...), which reads the module's serial number and uses the two least significant bytes, then triggers a channel 14 (temperature) ADC read and combines the two with some very basic math* to generate a sort of "multiplier seed" which can be used for randomness.

* (multiplication of the values by themselves)

https://bluegiga.zendesk.com/entries/59399217-Random-function
Smart lock

• Challenge-response, session key
• Commands encrypted by session key
• Challenge looks random
• Ranging: GPS-enabled, you have to leave the area and return
• What could possibly go wrong?
Smart lock - protocol

SESSION KEY = AES(Challenge, KEY)

Get "Challenge"

Challenge

Commands AES-encrypted by session key
Smart lock - protocol

SESSION KEY = AES(Challenge, KEY)

Get "Challenge"

Challenge

Close lock

OK, closed

MITM (intercept, record, pass through)
Smart lock – attack

SESSION KEY = AES(Challenge, KEY)

OK, closed (repeat the encrypted)

Close lock

Challenge (previously intercepted)

Get "Challenge"

The same as recorded session

MITM (replay)

OK, CLOSED!
Smart lock – DEMO
Smart lock – attack v2

Get "Challenge"

MITM
- Do not forward req to device.
- Advertise status "Closed"

OK, CLOSED!
Smart lock: AT commands

- BLE module AT interface exposed

### 7.2 Reset Commands

#### 7.2.1 Reset (ATRST)

<table>
<thead>
<tr>
<th>SD</th>
<th>RESET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function: Resets the module.</td>
<td></td>
</tr>
<tr>
<td>Command Format: ATRST</td>
<td></td>
</tr>
</tbody>
</table>

**Example(s):**

1. An ATRST is sent and once the module has reset, the RESET event is triggered.

**Command:** ATRST\(<cr>\\n**Response:** \(<cr>\_1f>\n\BR-LE4.0-S2<cr>\_1f>
## AT commands

**GET TEMPERATURE**

**Function:** Get the current temperature of the module’s internal temperature sensor.

**Command Format:** `ATT?`

**Response Format:** `<Temp_Celsius>,<Temp_Fahrenheit>`

**Response Value(s):**
- **Temp_Celsius:** Temperature in Celsius.
- **Temp_Fahrenheit:** Temperature in Fahrenheit.

**Example(s):**

```
COMMAND:   ATT?<cr>
RESPONSE:  <cr lf>
           OK
           <cr lf>
           026,079<cr lf>
```
7.8.2 UART Configuration (ATSUART)

**SET UART**

*Function:* Configures the module's UART. This command requires a reset for the new settings to take effect.

*Command Format:* ATSUART,<Baud_Rate>,<Parity>,<Stop_Bits>,<Flow_Control>

*Command Parameter(s):*
- **Baud_Rate:** 3-10 [9600bps – 1000000bps], enter Value from table below.
  (230400, 460800 and 1000000 are only available on Dual Mode modules.)

<table>
<thead>
<tr>
<th>Baud rate</th>
<th>Value</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9600</td>
<td>3</td>
<td>0.14</td>
</tr>
<tr>
<td>19200</td>
<td>4</td>
<td>0.14</td>
</tr>
<tr>
<td>38400</td>
<td>5</td>
<td>0.14</td>
</tr>
<tr>
<td>57600</td>
<td>6</td>
<td>0.03</td>
</tr>
<tr>
<td>115200</td>
<td>7</td>
<td>0.03</td>
</tr>
<tr>
<td><strong>230400</strong></td>
<td><strong>8</strong></td>
<td><strong>0.03</strong></td>
</tr>
<tr>
<td>460800</td>
<td>9</td>
<td>0.03</td>
</tr>
<tr>
<td><strong>1000000</strong></td>
<td><strong>10</strong></td>
<td><strong>0.03</strong></td>
</tr>
</tbody>
</table>
AT commands

7.8.3 PIO Configuration (ATSPIO)

SET PIO

**Warning:** Applying an external voltage to a PIO assigned as an output may permanently damage the module. The maximum voltage level on any pin should not exceed 3.6V. The I/O is NOT 5V tolerant.

**Function:** Sets the direction and values of PIO’s.

**Command Format:** ATSPIO,<PIO_Num>,<Direction>,<Value>

**Command Parameter(s):**
- **PIO_Num:**
  - Single Mode: 0,1,2,5,7,8,9,10,11,12,13,14
  - Dual Mode: 0,1,2,5,7,8,9,10,11,12,13,14,19,20,21,22
Fallback to analog key may be unavailable...

No more keys!
DEMO: AT commands

sent CMD: ATSCL?
OK
0
ATSUART?
Switch to CMD mode
sent CMD: ATSUART?
OK
3,0,0,0
ATT?
Switch to CMD mode
sent CMD: ATT?
OK
027,081
ATSN?
Switch to CMD mode
sent CMD: ATSN?
OK
LockECFE7E139F95
DEMO: Anti-thief
DEMO: interception – static password
DEMO: Mobile PoS
But what about BLE encryption?
"Just Works"

No need for bonding

Other MAC

MITM

Bond – encrypted communication

(for static attack scenarios not necessary)
"Just Works"

Bond – encrypted communication

Cloned MAC

MITM

Bond – encrypted communication

(for static attack scenarios not necessary)
Bluetooth smart?

Have you tried turning it off and on again?
Remove the pairing, now it works again!
PIN entry – trick into pairing again, sniff, crack
PIN entry – trick into pairing again, sniff, crack
Some attacks

• Denial of Service
• Interception
• Replay
• Authentication bypass
• Proximity actions
• Misconfiguration/excessive services abuse
• Logic flaws
• Badly designed crypto
• Brute force
• Fuzzing
• ...

How to fix the problem?

• Use the BLE security features
  • Encryption, bonding, MAC randomization
  • Do not allow to bond automatically
  • Detect MITM, warn the user
• Your own mechanisms
  • Do not implement static passwords
  • Design with active interception possibility in mind
• Beware excessive services, misconfiguration
• Prepare fallback for Denial of Service
• ...
• More details in whitepaper
Q&A?

More information, these slides, whitepaper, tool source code:

slawomir.jasek@securing.pl  @slawekja
Thanks:

• My family – for patience and various favours

• SecuRing – for funding large part of this research