BLESA: Spoofing Attacks against Reconnections in Bluetooth Low Energy

Jianliang Wu¹, Yuhong Nan¹, Vireshwar Kumar¹, Dave (Jing) Tian¹, Antonio Bianchi¹, Mathias Payer², Dongyan Xu¹

¹ Purdue University ² EPFL
Motivation

- Bluetooth Low Energy (BLE) devices are ubiquitous
  - Smart home devices
    - Smart temperature sensor
  - Health care devices
    - Smart glucose monitor

Over 5 billion BLE enabled devices
Motivation

- **BLE security mechanism**
  - **Security level**
    - Level 1
      - No security
    - Level 2
      - Encryption
    - Level 3 and 4
      - Encryption and authentication
  - **Bluetooth pairing**
    - No I/O interfaces
      - Level 2 (unauthenticated key)
    - With I/O interfaces
      - Level 3 and 4 (authenticated key)
Motivation

- BLE security mechanism
  - Server-client architecture
    - BLE uses request and response scheme
    - Data is stored as attribute on server device
    - Each attribute has security requirements
  - Server-side security enforcement
    - Server checks whether the current security level match the requirement or not

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Security Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Name</td>
<td>“Oura Ring”</td>
<td>Level 1</td>
</tr>
<tr>
<td>Battery level</td>
<td>“90%”</td>
<td>Level 2</td>
</tr>
</tbody>
</table>
Motivation

• Attacks on BLE
  ▪ Eavesdropping[^1]
  ▪ Illegal access by compromising client BLE device[^2]
    ○ Reading glucose level
    ○ Opening smart lock
  ▪ Man-In-The-Middle Attacks against *unpaired* BLE devices[^3]
    ○ Manipulating user data

Motivation

• Prior attacks on BLE
  ▪ Some attacks target the pairing procedure for **first-connection** and **unpaired devices** [WOOT’13, blackhat’16]
  ▪ Some other attacks need **additional assistance** [NDSS’14, SEC’19, NDSS’19]
    ○ Malicious app on the phone

• Unexplored reconnection procedure
Our Work

• Formal analysis of BLE *reconnection* procedure
  ▪ Two design weaknesses identified

• BLE Spoofing Attacks (BLESAs) against *paired* devices *without* extra assistance
  ▪ Do not need malicious apps

• Evaluation on real-world BLE devices
  ▪ Affecting more than 1 billion real-world BLE devices and 16,000 BLE apps
Formal Analysis and Findings

• Formal model
  ▪ Modeling BLE reconnection procedure using ProVerif
  ▪ Verifying security properties
    ○ Confidentiality, Integrity, and Authenticity

• Identified Weaknesses
  ▪ Optional authentication
  ▪ Circumventing authentication
    ○ Reactive authentication
      ❖ Design issue
    ○ Proactive authentication
      ❖ Implementation issue

BLE Spoofing Attacks (BLESAs)
BLESA against Reactive Authentication

Reactive authentication

Client

Reconnect to a paired server device

Connected

Enable encryption

Accept attribute value

Server

Connection request

(Plaintext, level 1)

Connected

Enable encryption

Request (battery level)

(Plaintext, level 1)

Insufficient Encryption

(Plaintext, level 1)

Request (battery level)

(Encrypted, level 2)

Response ("90%")

(Encrypted, level 2)

Accept attribute value

Attack reactive authentication

Client

Reconnect to a paired server device

Connected

Enable encryption

Request (battery level)

(Plaintext, level 1)

Level 2 needed

Adversary

Advertise as benign server

Level 1 needed

Request (battery level)

(Plaintext, level 1)

Spoofed value ("0%")

(Plaintext, level 1)

Level 2 needed

Attribute | Value | Security Requirement
--- | --- | ---
Battery level | "90%" | Level 2
BLESA against Proactive Authentication

Proactive authentication

Client
- Connected
- Encrypted
- Request (battery level) (Encrypted, level 2)
- Response (“90%”) (Encrypted, level 2)
- Accept attribute value

Server
- Connected
- Encrypted
- Enable encryption
- Level 2 needed

Attack proactive authentication

Client
- Connected
- Encrypted
- Enable encryption

Server
- Connected
- Encrypted
- Enable encryption
- Connected

Adversary
- Advertise as benign device
- Level 1 needed
- No key

Connection request
- Level 2 needed
- Connected
- Encryption fails
- Accept spoofed attribute value
- Request (battery level) (Plaintext, level 1)
- Spoofed value (“0%”)
- Accept spoofed attribute value
- Connected
- Connection continues in PLAINTEXT
Evaluation and Impact

• Weakness 1 (optional authentication) examination
  ▪ Whether the BLE apps use authentication during reconnection?
  ▪ Whether the real-world server BLE devices use authentication during reconnection?

• Weakness 2 (circumventing authentication) examination
  ▪ Which authentication procedure is during reconnection used by main-stream BLE stacks?
  ▪ Whether the used authentication procedure is vulnerable to BLESAA?
Evaluation and Impact

• Weakness 1 (optional authentication)
  ▪ Whether the BLE apps use authentication during reconnection?
    ○ Analyzing BLE apps
    ○ 86/127 (67.7%) of analyzed BLE apps do not use authentication during reconnection
  ▪ Whether the real-world server BLE devices use authentication during reconnection?
    ○ Analyzing real-world server BLE devices
    ○ 10/12 of analyzed BLE devices do not support authentication during reconnection

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Auth.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nest Protect Smoke Detector</td>
<td>×</td>
</tr>
<tr>
<td>Nest Cam Indoor Camera</td>
<td>×</td>
</tr>
<tr>
<td>SensorPush Temperature Sensor</td>
<td>×</td>
</tr>
<tr>
<td>Tahmo Tempi Temperature Sensor</td>
<td>×</td>
</tr>
<tr>
<td>August Smart Lock</td>
<td>×</td>
</tr>
<tr>
<td>Eve Door &amp; Window Sensor</td>
<td>×</td>
</tr>
<tr>
<td>Eve Button Remote Control</td>
<td>×</td>
</tr>
<tr>
<td>Eve Energy Socket</td>
<td>×</td>
</tr>
<tr>
<td>Ilumi Smart Light Bulb</td>
<td>×</td>
</tr>
<tr>
<td>Polar H7 Heart Rate Sensor</td>
<td>×</td>
</tr>
<tr>
<td>Fitbit Versa Smartwatch</td>
<td>√</td>
</tr>
<tr>
<td>Oura Smart Ring</td>
<td>√</td>
</tr>
</tbody>
</table>
Evaluation and Impact

• Weakness 2 (circumventing authentication)
  ▪ Which authentication procedure is used for main-stream BLE stacks?
  ▪ Whether the authentication procedure is vulnerable to BLESAA?
    ○ Analyzing main-stream BLE stacks

<table>
<thead>
<tr>
<th>Platform</th>
<th>OS</th>
<th>BLE Stack</th>
<th>Authentication</th>
<th>Issue</th>
<th>Vulnerable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux Laptop</td>
<td>Ubuntu 18.04</td>
<td>BlueZ 5.48</td>
<td>Reactive</td>
<td>Design</td>
<td>Yes</td>
</tr>
<tr>
<td>Google Pixel XL</td>
<td>Android 8.1, 9, 10</td>
<td>Fluoride</td>
<td>Proactive</td>
<td>Implementation</td>
<td>Yes</td>
</tr>
<tr>
<td>iPhone 8</td>
<td>iOS 12.1, 12.4, 13.3.1</td>
<td>iOS BLE stack</td>
<td>Proactive</td>
<td>Implementation</td>
<td>Yes</td>
</tr>
<tr>
<td>Thinkpad X1 Yoga</td>
<td>Windows 10 V. 1809</td>
<td>Windows stack</td>
<td>Proactive</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>
Evaluation and Impact

BLESA against Oura Ring Demo
Evaluation and Impact

• Impact
  ▪ Affected BLE apps
    ○ At least 8,000 Android BLE apps with 2.38 billion installations\(^1\)
    ○ Similar number may apply to iOS apps
  ▪ Affected server BLE devices
    ○ More than 1 billion BLE devices\(^1\)
  ▪ Medeia report
    ○ Security Boulevard

Evaluation and Impact

• Responsible disclosure
  ▪ Apple Product Security
    ○ CVE-2020-9770
  ▪ Android Security Team
    ○ Reported on April 8, 2019

The Android Security Team believes that this is a duplicate of a report previously submitted by another external researcher on Apr 5, 2019.

The duplicate issue is being tracked by AndroidID-130833727.

Thank you,
Android Security Team
Mitigations

• Reactive authentication
  ▪ Updating specification
    ○ Removing reactive authentication
    ○ Exchanging attributes’ security requirements during pairing

• Proactive authentication
  ▪ Fixing vulnerable implementations
    ○ iOS BLE stack
      ❖ Apple issued iOS 13.4 and iPadOS 13.4 to fix the vulnerability
    ○ Android BLE stack (Fluoride)
    ○ Linux BLE stack (BlueZ)
      ❖ Changing to proactive authentication
Summary

• Formal analysis of the BLE reconnection procedure
• BLESA against paired BLE devices
• Evaluation on real-world BLE devices

Thank you! Questions?
This work was supported in part by ONR under Grant No0014-18-1-2674.
wu1220@purdue.edu