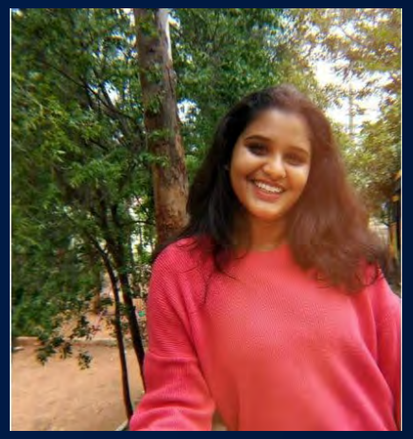
DEEP
ARMOR

Attacking Bluetooth LE Design and Implementation in Mobile + Wearables Ecosystems

Meghana Bidarahalli and Ananya.M.Gowda

Speakers

- Meghana Bidarahalli
- Security Analyst

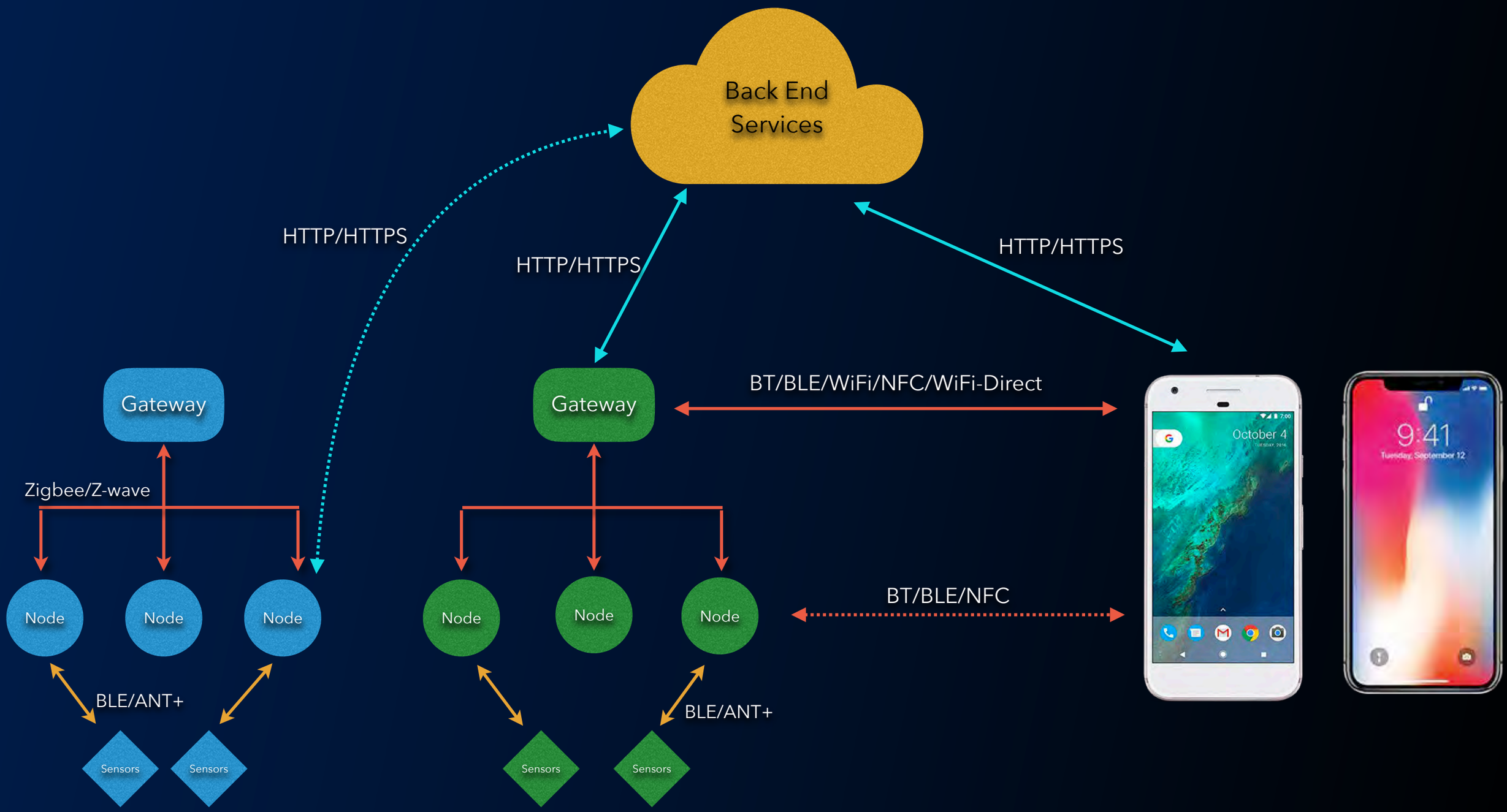


- Ananya M Gowda
- Security Analyst

Agenda

- Blueprint of an IoT/wearable ecosystem
- Challenges: Securing a modern-day gadget
- Introduction to Bluetooth & BLE Security
- Attacking Bluetooth and BLE networks
  - IoT - Android/iOS ecosystems [Demo]
- Recommendations for Ecosystem Security
- Summary

IoT/Wearable Ecosystem



Case Study: Fitness Trackers

Wearable = Comfortable

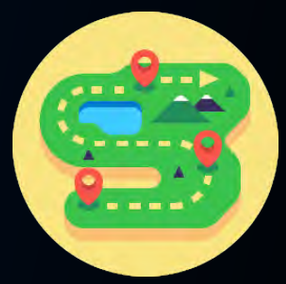
Smart

Untethered

Continuous Learning

Data/Analytics

Better Quality of Life



Challenges: Securing a modern-day gadget

- Rapid time-to-market
- Constantly evolving requirements
- Diverse, non-standard and evolving communication protocols
- Known security weaknesses
- Long lives for IoT products
- Privacy
- Nascent research in IoT security

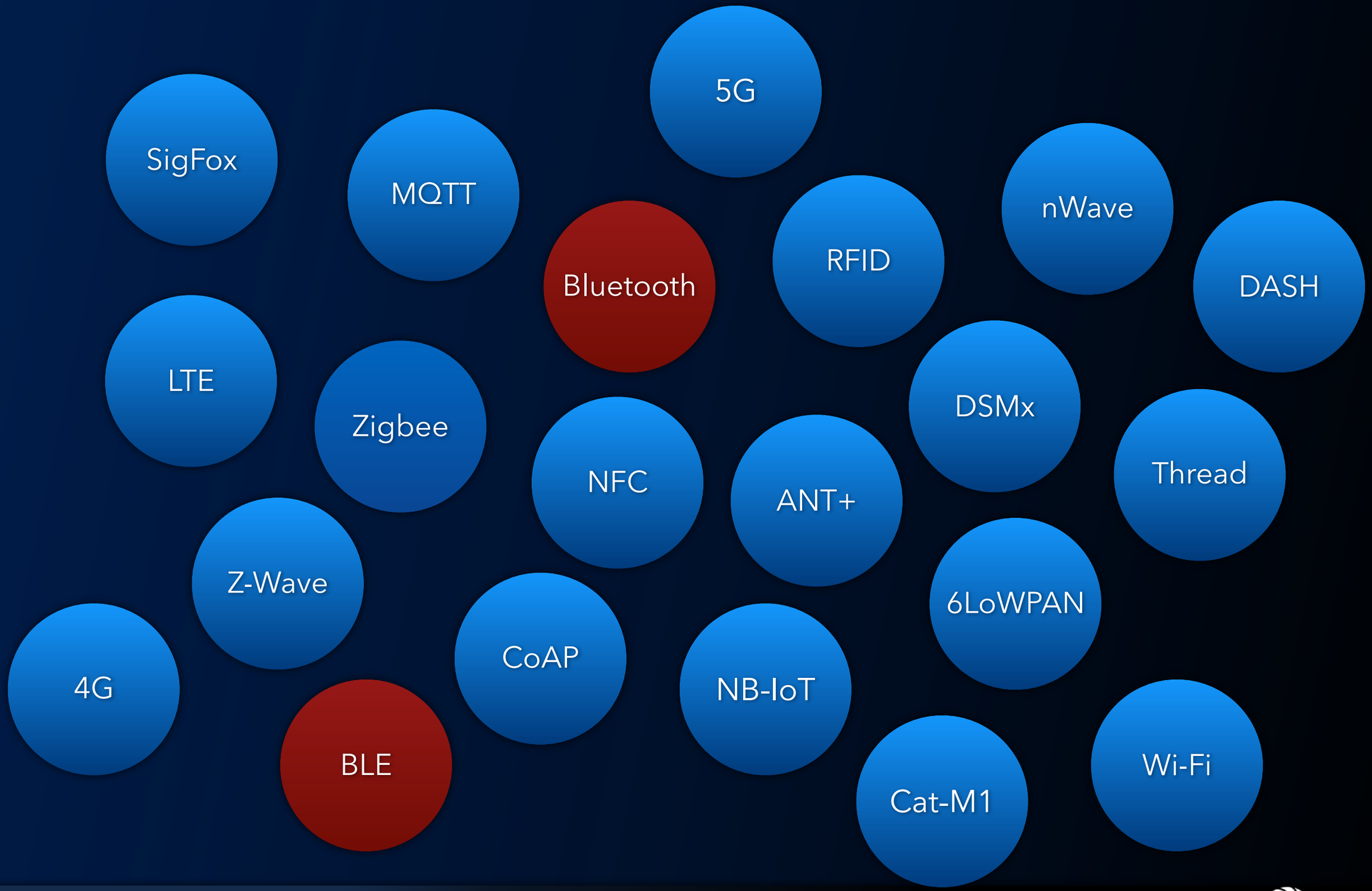
Challenges - Technical

- Collection of personal data and PII is higher
  - Geo-location information
  - Biometric data
  - Sensor data
  - Payment services

- Limited SW stack –> security may get compromised
  - Often FW running on micro-controllers
  - Field updates are difficult
  - Asymmetric key crypto, TEEs, etc. are heavy

- Multi-tier, multi-tenant product architecture
  - Cross-domain flows
  - Multiple exposure points as a consequence

Today's Agenda



BLE Introduction

- Wireless protocol for short range data exchange
  - BT: 1-100m; BLE: 10-600m
- BLE = Light-weight subset of classic Bluetooth with low power consumption
- RF range: 2.4 - 2.485 GHz
- Maintained & Governed by the Bluetooth Special Interest Group (SIG)
- Popular use cases: wearable devices, smart pay systems, healthcare, smart security systems etc

Bluetooth LE security

Secure Simple Pairing (SSP)

- Just Works: very limited/no user interface
- Numeric Comparison: devices with display or yes/no button
- Passkey Entry: 6 digit pin as the pass key
- Out Of Band: Out of the band channel for key exchange to thwart MITM attacks

- **Network traffic is encrypted with AES-128**

Known weaknesses in BT/BLE

- Security of the communication link depends on pairing algorithm
- Eavesdropping on pairing mechanism compromises encryption keys
- ‘Just works’ mode prone to MITM attacks
- **Apps on the phone**

Problem: Ecosystem

Fitness Trackers

- Sports/Activity Band Products
- Social Fitness
- Many market wearables are affected
  - Popular fitness tracker Responsibly Disclosed
- Classic example of an ecosystem problem

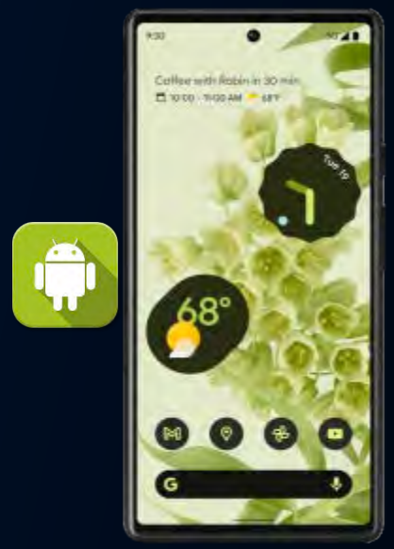
Ecosystem overview

Device communication

Device Commands:
- Put device into recovery mode
- Do a FW update
- Change Device (BLE) name

Notifications:
- Social apps
- Calls and texts

Information:
- User activity data
- User profile updates
- Application action (calls, music control)
- Call/text/social updates (sometimes)



The Problem – Prelude

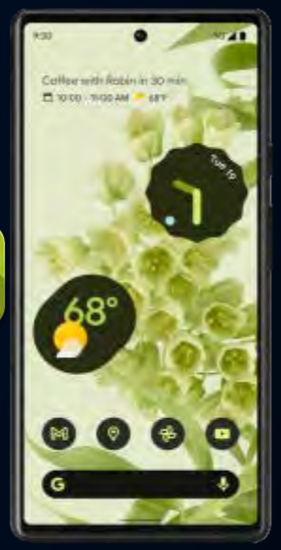
Device Commands:
- Put device into recovery mode
- Do a FW update
- Change Device (BLE) name

Notifications:
- Social apps
- Calls and texts

Information:
- User activity data
- User profile updates
- Application action (calls, music control)
- Call/text/social updates (sometimes)

BLE - ENCRYPTED

ATTACKER



The Problem

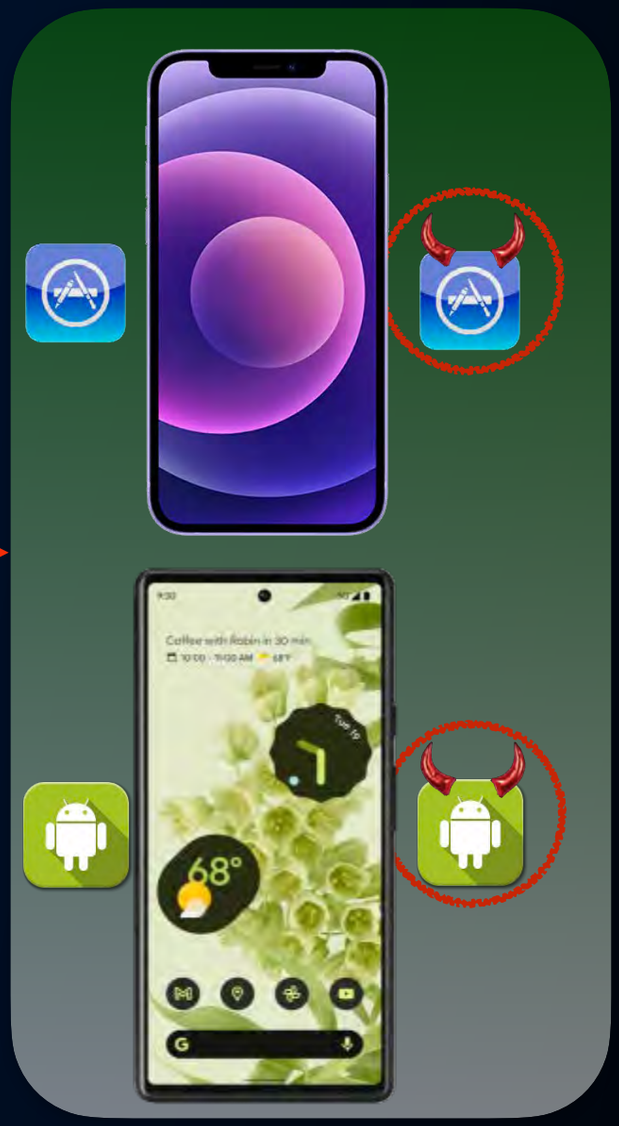
BLE - ENCRYPTED

ATTACKER

Device Commands:
- Put device into recovery mode
- Do a FW update
- Change Device (BLE) name

Notifications:
- Social apps
- Calls and texts

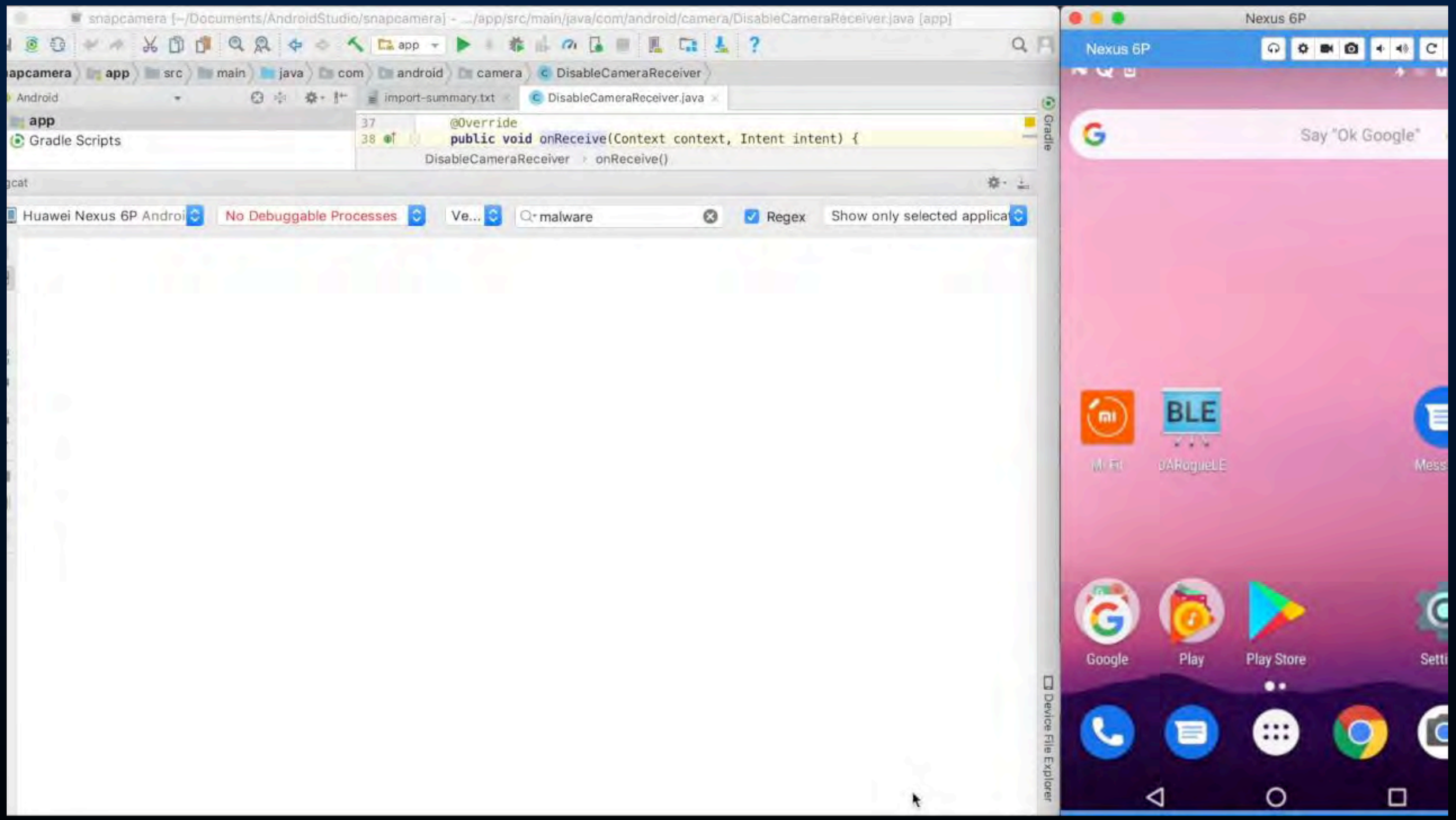
Information:
- User activity data
- User profile updates
- Application action (calls, music control)
- Call/text/social updates (sometimes)



Demo - Setup

- Xiaomi Mi Band 2 (FW v1.0.1.81)
- Smartphone running latest Android
- Xiaomi Mi Fit App v5.5.2
- Deep Armor’s custom malware app

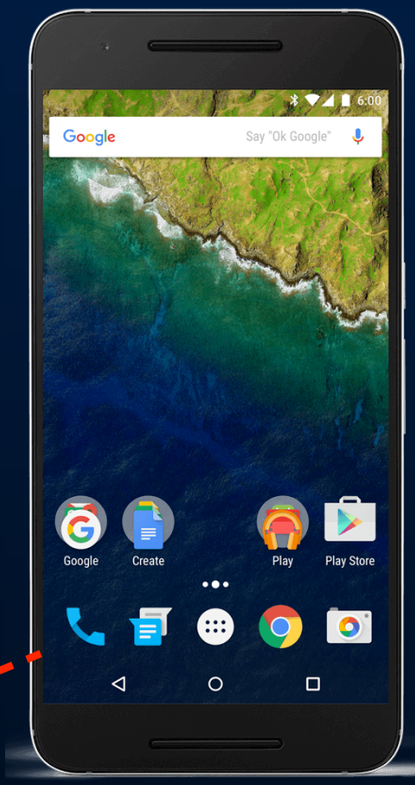
Instructor Demo



Instructor Demo

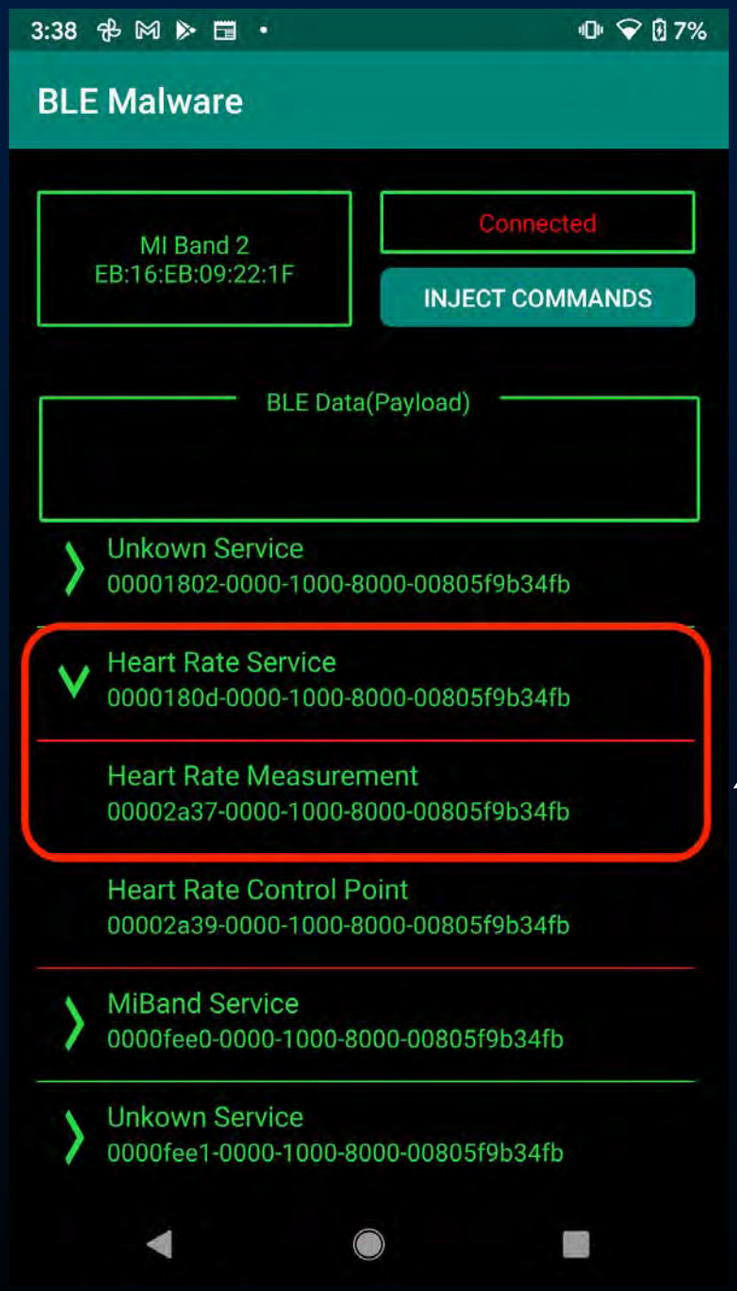
Activity Data, Notifications, etc.

Commands, FOTA, etc.



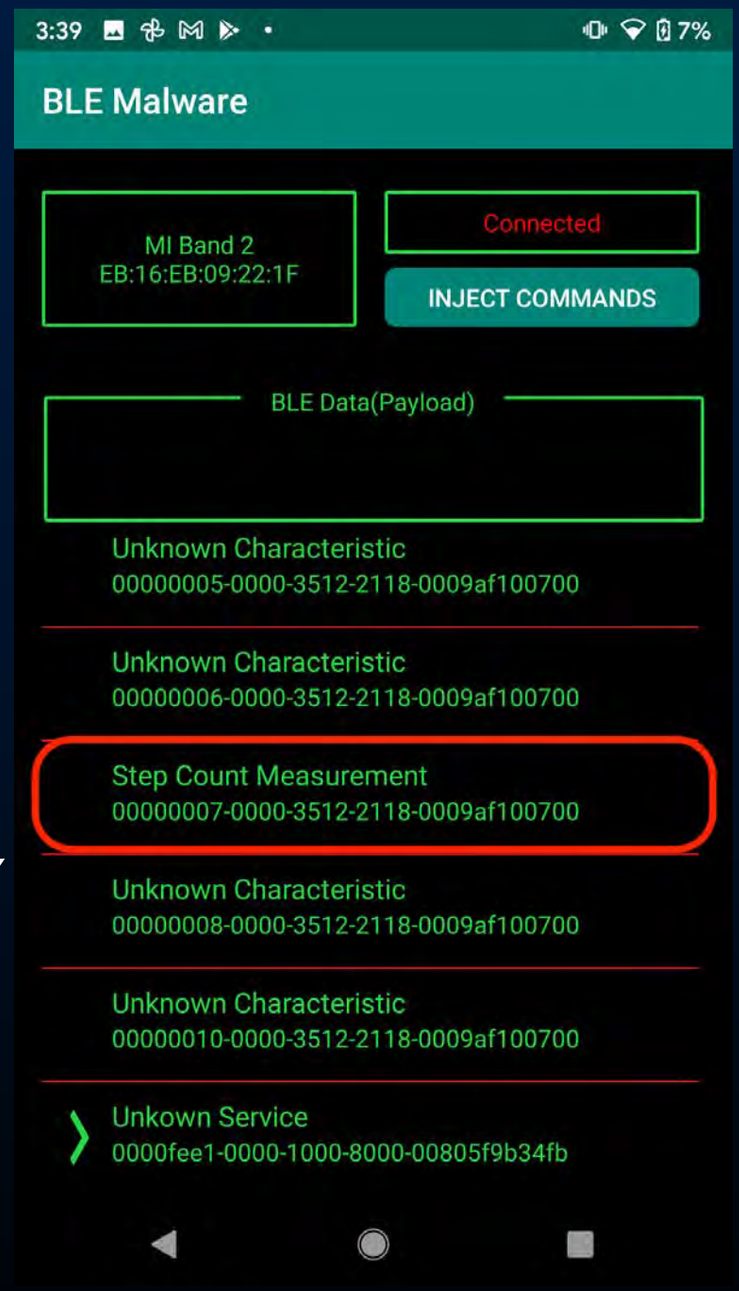
Step Count & Calories

GATT Profile



UUID to fetch heart rate

UUID to fetch user steps



Root Cause

Any app on Android/iOS can read/write data on the BT/BLE channels (just like the legitimate app)

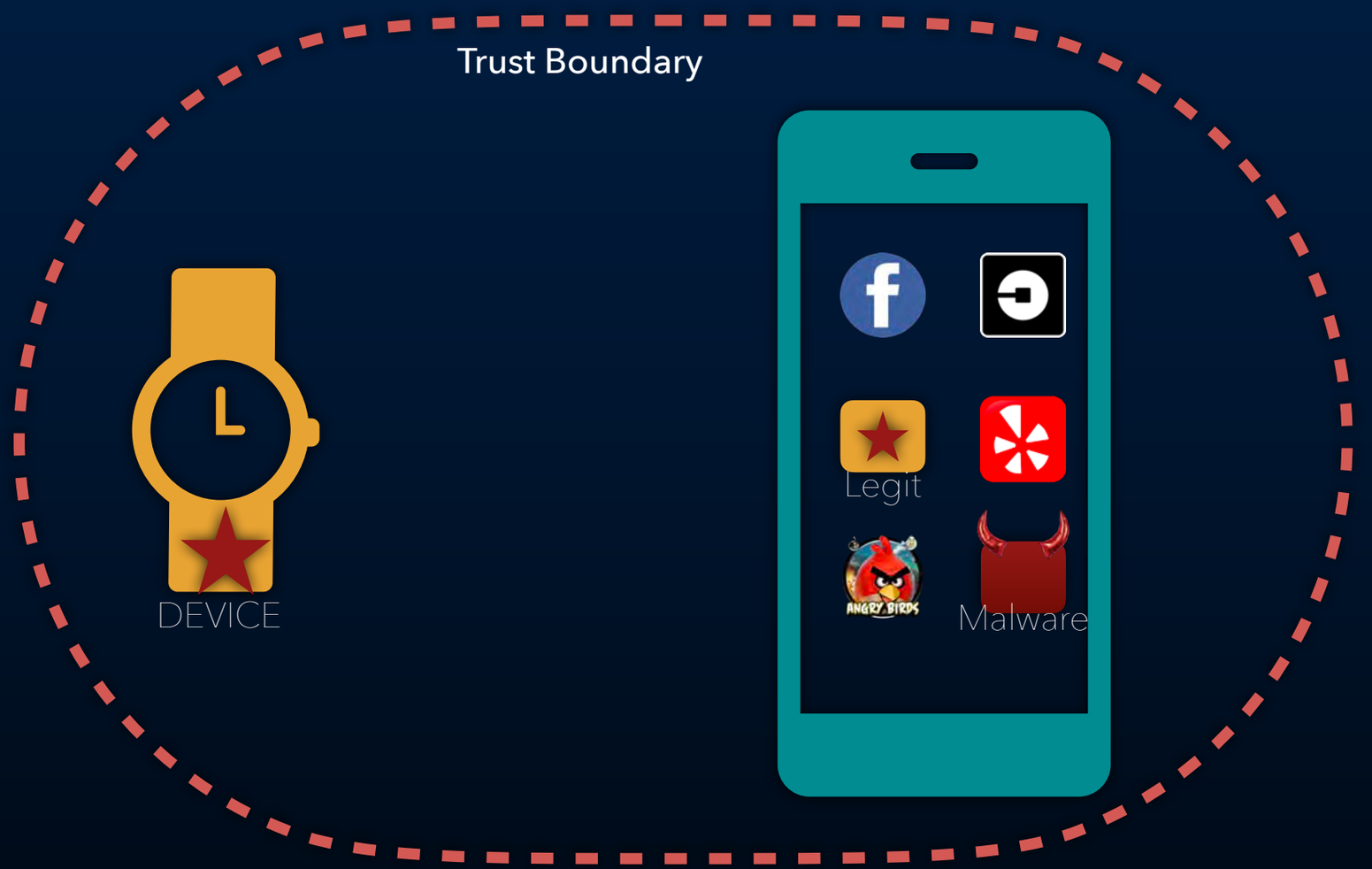
- Android
  - android.permission.BLUETOOTH
  - android.permission.BLUETOOTH_ADMIN – quote:

> If you want your app to initiate device discovery or manipulate Bluetooth settings, you must declare the `BLUETOOTH_ADMIN` permission. Most apps need this permission solely for the ability to discover local Bluetooth devices. Don't use the other abilities granted by this permission unless the app is a "power manager" that modifies

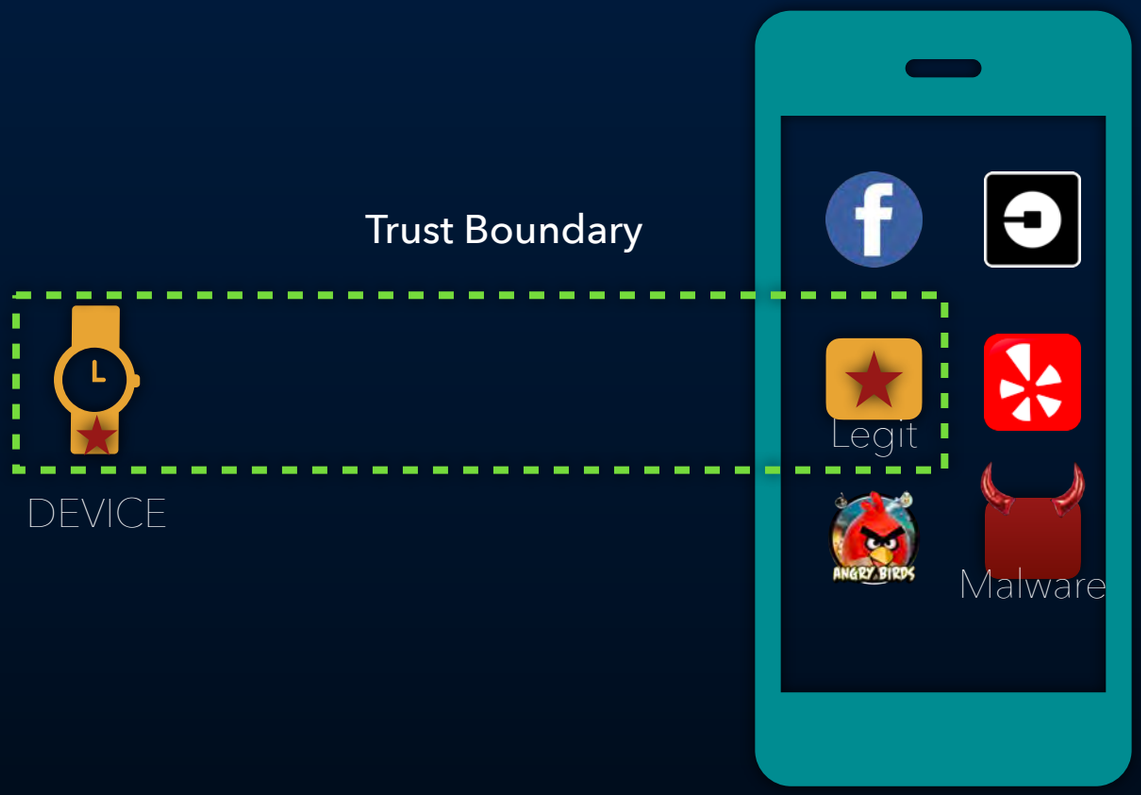
> **Caution:** When a user pairs their device with another device using BLE, the data that's communicated between the two devices is accessible to **all** apps on the user's device.

- iOS
  - Core Bluetooth (CB) Framework
  - Centrals (client/phone) and Peripherals (server/wearable) classes

Problem - Trust Model



Solution - Trust Model



Mitigation


Mobile Device
BLE Link Layer
Malware App
Legit App
Encyrption
App specific key

Summary

- Next-gen SDLC
  - IoT Security = device + mobile + cloud + wireless
  - Continuous Security for CI/CD

- Security, Privacy and Legal woven into the development cycle

- Leveraging industry standards

Ecosystem | Protocols | Integration | Interoperability

DEEP ARMOR

SDLC

Vulnerability Assessments

Security Consulting

Trainings

www.deeparmor.com | @deep_armor | services@deeparmor.com