TU Graz

SCIENCE
PASSION
TECHNOLOGY

AndroGUARD: Implementing fingerprinting mitigation on Android

Gergö Kranz

05.07.2024

Outline

1. Motivation
2. Fingerprinting
3. Main Question
4. Background on existing work
5. Our Approach
6. Deployment
7. Validation

Motivation

How does Fingerprinting work

- Extract information about the system
- Analyse extracted information
- Combine unique values into a unique identifier

How does Fingerprinting work

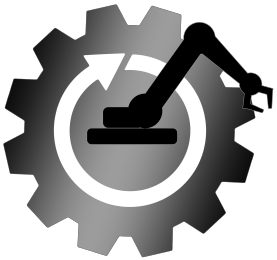
- Extract information about the system
- Analyse extracted information
- Combine unique values into a unique identifier

How does Fingerprinting work

- Extract information about the system
- Analyse extracted information
- Combine unique values into a unique identifier

Fingerprinting Sensors

- Builtin Error
- Is consistent over the lifetime of the sensor
- Hardware can be uniquely identified



Fingerprinting Sensors

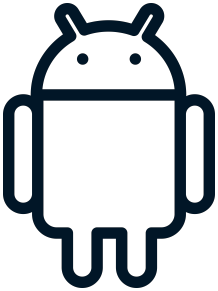
- Records multiple measurements
- Calculates the deviation from expected values

Main Question

How to protect against fingerprinting

Existing Methods

- Access Control



Related Works

- Many papers are looking into exploiting sensor fingerprinting
- Do not implement a solution for the Android API

Related Works

- Many papers are looking into exploiting sensor fingerprinting
- Do not implement a solution for the Android API

Proposed Solutions

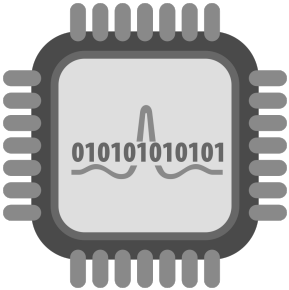
- Recalibrate the sensors to be perfect
  - Gets rid of the error
  - Can not be done easily

Proposed Solutions

- Add noise to conceal the calibration errors
  - Apply a random offset and gain to each measurement
  - Can be done without any user interaction

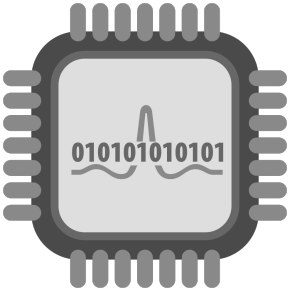
Protecting Sensors

- Mask builtin error by added noise
- Use the A2P2 framework for deployment



Protecting Sensors

- Mask builtin error by added noise
- Use the A2P2 framework for deployment



Implementation

- Intercept the original method
- Apply appropriate random noise
- Return obstructed sensor data to original method

Implementation

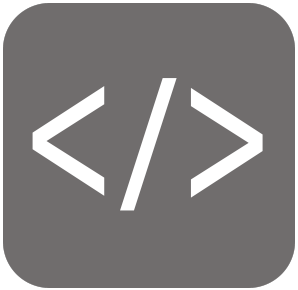
- Intercept the original method
- Apply appropriate random noise
- Return obstructed sensor data to original method

Implementation

- Intercept the original method
- Apply appropriate random noise
- Return obstructed sensor data to original method

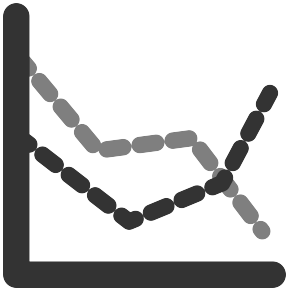
A2P2

- Incorporates the patch into a valid APK
- Intercepts the original function calls
- Executes the patch



Validation

- Comparing values before and after the patch
- Could not be done sufficiently due to limited access to supported hardware



Conclusion

- Masking the sensor values decreases fingerprintability
- Modifying the SensorEventListener makes it easy to incorporate the patch into the Android API