6thSense

Context-aware Sensor-Based Attack Detector for Smart Devices

Usenix Security Symposium 2017

Amit Kumar Sikder, Hidayet Aksu, and A. Selcuk Uluagac.

GSD Meeting 30-11-2018 - Rui Claro

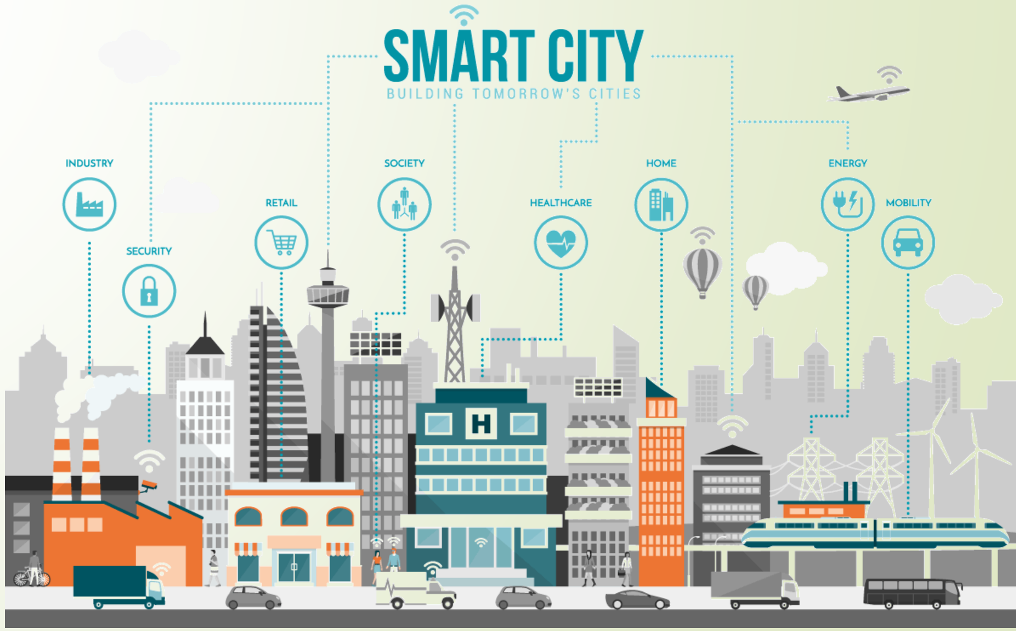
Outline

- Introduction
- Technical Approach
- Performance Overview
- Conclusions and Future Work
- Discussion

Introduction

Background

- Smartphones
- Wearables
- Smart Homes
- Smart Cities



New Sensor-based Threats

- Eavesdropping
- Keystroke Inference
- Location Inference
- Triggering Malware

Motivation

- Users are not knowledgeable about the threats
- Users are unaware of the consequences
- Rapid growth of threats in recent years
- Failure of existing sensor management systems

Contributions

- Sensor-based Attack Detector
  - 6thSense
- Real-life user data
  - From 50 Users
- High detection rate
  - Small system overhead

Technical Approach

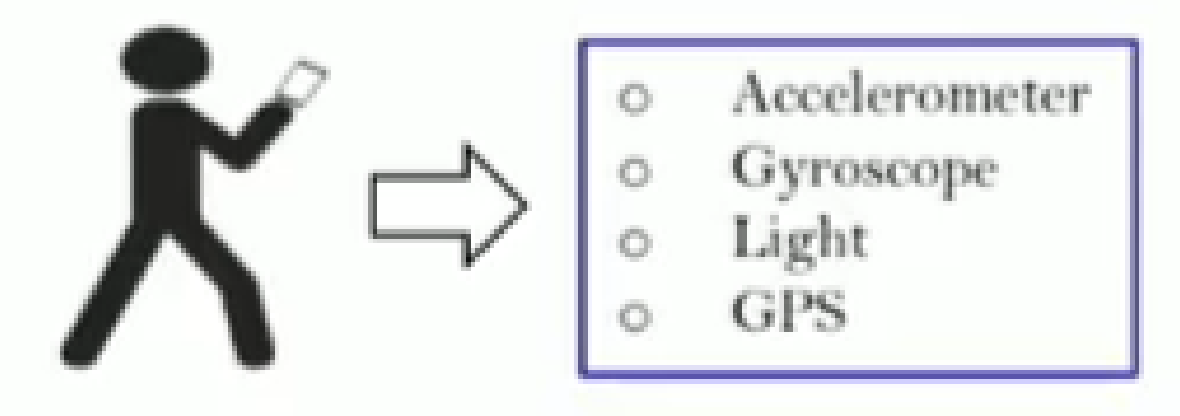
Existing Sensor Management Systems

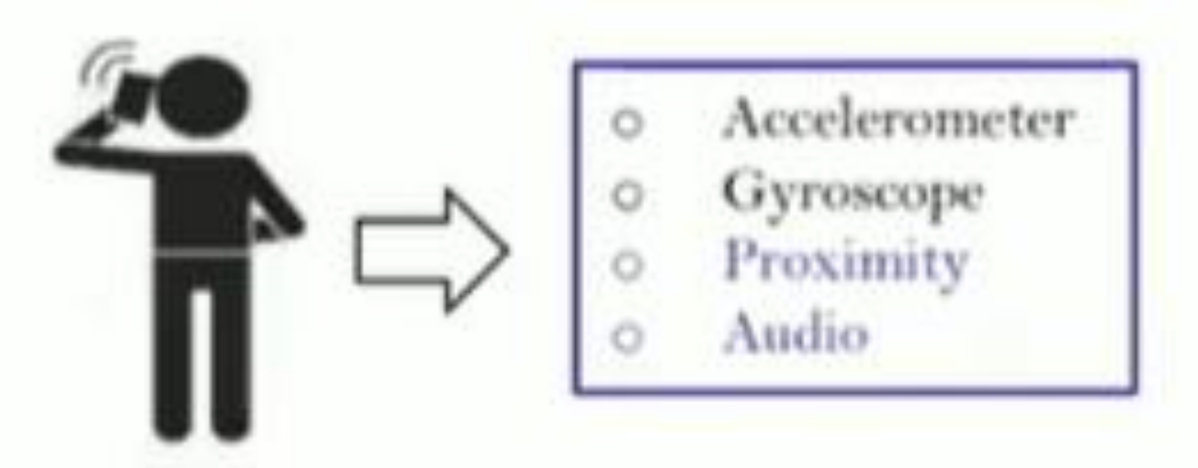
- Similar sensor management frameworks for existing operating systems (e.g., Android, iOS).
- Permission-based access only.
  - Only selected sensors are covered.
- No permission for accessing other sensors
  - E.g., accelerometer, light sensor, etc.
- No user control over sensor after granting permission.
- No subsequent knowledge for users about information accessed via sensors.

Threat Model

- Stealing Information via Sensor
  - Exploiting sensors to capture information on a device and reveal them to an attacker.
- Triggering Malware via Sensor
  - Malicious app installed in the device triggered by a message via sensors.
- Information Leakage via Sensor
  - Information saved or recorded in the device transferred via sensor.

Context Awareness

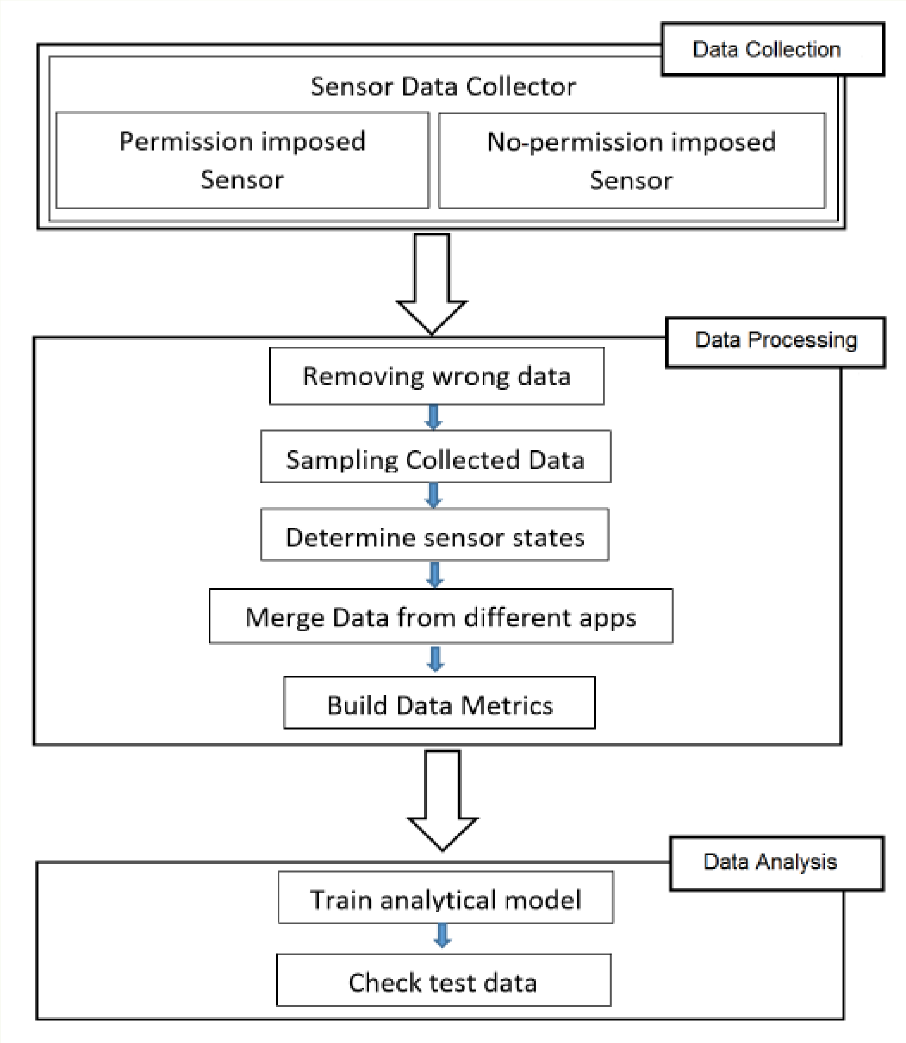




Sensor Co-dependence

- For each user task/activity, a **specific set of sensors** remains active.
- Sensors are considered as co-dependent entities for each task/activity.
- By observing which sensors are active for a task/activity, it is possible to differentiate between benign activities and malicious activities.

6thSense: Framework Overview



6thSense: Detection Techniques

- Markov Chain
- Naïve Bayes
- Other Machine Learning Techniques
  - Logistic Function, J48, etc.

Performance Evaluation

Performance Evaluation

- Data collected from 50 different users.
- Nine tasks/activities selected.

Task Category	**Task Name**
Generic Activities	1. Sleeping
	2. Driving as driver
	3. Driving as passenger
User-related Activities	1. Walking with phone in hand
	2. Walking with phone in pocket/bag
	3. Playing games
	4. Browsing
	5. Making phone calls
	6. Making video calls

- 75% of data used for training, 25% of data used for test.
- Performance Metrics:
  - Recall Rate, ROC, PRC, Accuracy, F-Score, etc.

Markov Chain Based Detection Results

Threshold (Number of consecutive malicious states)	Recall rate	False negative rate	Precision rate (specificity)	False positive rate	Accuracy	F-score
0	0.62	0.38	1	0	0.6833	0.7654
1	0.86	0.14	1	0	0.8833	0.9247
2	0.96	0.04	1	0	0.9667	0.9796
3	0.98	0.02	1	0	0.9833	0.9899
5	1	0	0.9	0.1	0.9833	0.9474
6	1	0	0.8	0.2	0.9667	0.8889
8	1	0	0.6	0.4	0.9333	0.75
10	1	0	0.5	0.5	0.9167	0.6667
12	1	0	0.5	0.5	0.9167	0.6667
15	1	0	0.3	0.7	0.8833	0.4615

Naïve Bayes Model Results

Threshold Probability	Recall rate	False negative rate	Precision rate (specificity)	False positive rate	Accuracy	F-score
55%	1	0	0.6	0.4	0.9333	0.75
57%	1	0	0.7	0.3	0.95	0.8235
60%	1	0	0.7	0.3	0.95	0.8235
62%	1	0	0.7	0.3	0.95	0.8235
65%	0.94	0.06	0.7	0.3	0.9	0.8024
67%	0.88	0.12	0.7	0.3	0.85	0.7797
70%	0.7	0.3	0.8	0.2	0.7167	0.7467
72%	0.7	0.3	0.9	0.1	0.7333	0.7875
75%	0.66	0.34	0.9	0.1	0.7	0.7616
80%	0.66	0.34	0.9	0.1	0.7	0.7615

Detection with other Machine Learning approaches

Algorithms	Recall rate	False negative rate	Precision rate	False positive rate	Accuracy	F-score
PART	0.9998	0.0002	0.6528	0.3472	0.99	0.7899
Logistic Function	0.9997	0.0003	0.2778	0.7222	0.998	0.4348
J48	0.9998	0.0002	0.6528	0.3472	0.99	0.7899
LMT	0.9998	0.0002	0.9306	0.0694	0.9997	0.964
Hoeffding Tree	1	0	0.0556	0.9444	0.9978	0.1053
Multilayer Perceptron	0.9998	0.0002	0.6944	0.3056	0.9991	0.8196

Performance Overhead

Parameters	Time	No-permission imposed sensors	Permission imposed sensors
CPU Usage	N/A	3.90%	0.3%
RAM Usage	N/A	23 MB	14 MB
Disc Usage	For 1 min	6.5 MB	1 KB
	For 5 min	9 MB	2 KB
	For 10 min	12 MB	3 KB
Power Consumption	1 min	13.5 mW	3.12 mW
	5 min	96.67 mW	27.4 mW
	10 min	133.33 mW	45 mW
Power Consumption (without datafile)	1 min	2.68 mW	0.23 mW
	5 min	23.4 mW	9.63 mW
	10 min	55.35 mW	17 mW

Conclusions and Future Work

- Contributions
  - Novel context-aware sensor-based attack detector.
  - Machine Learning techniques used to develop the framework.
  - Evaluation based on data collected from real users.
  - High detection rate with minimum system overhead.
- Future Work
  - Implement the framework for small handheld devices such as fitness bands.

Discussion

- Prototype of 6thSense developed only for Samsung Galaxy s5 Duo.
  - Sensors have different specification for different devices
  - Reimplementation needed for other devices
- Machine Learning training is done offline.
  - Training could be outsourced to the cloud
    - Privacy concerns in transferring sensor data
- Collection of data done in a compromised device.
  - Tainted data for training

6thSense and my thesis

- Broad thesis topic:
  - Privacy and Security in the Internet of Things
- Currently working on my TI (Tópicos de Investigação) course:
  - Intrusion Detection Systems, Machine Learning
- Future Work of 6thSense is a possible path
  - Expand to a distributed cloud based solution
  - Using privacy-preserving techniques