Department of Information & Communication Systems Engineering
University of the Aegean
Karlovasi, Samos, Greece

A cloud-based architecture to crowdsource mobile app privacy leaks

Dimitrios Papamartzivanos
dpapamartz@aegean.gr

Dimitrios Damopoulos
ddamopou@stevens.edu

Georgios Kambourakis
gkamb@aegean.gr

Outline

- Introduction
- Motivation
- Current Solutions – Related Work
- Our contribution
- Our proposal
- Evaluation
- Conclusion

Introduction

- Mobile devices have become an integral part of people's life.
  - Resulting in attracting the attention of ill-motivated entities.
  - Malicious apps aim either to expose user privacy or manipulate services and data stored on the device.
- Android holds 75% of the total mobile market share.
  - 98% of malicious apps in the wild target to Android.

Motivation

- Android markets do not perform any privacy-exposure related control.
- Apps, malicious or not, ask for many privacy-sensitive permissions not respecting users' privacy.
- External (along with internal) measures for safeguarding the privacy of the end-user are needed.

Current Solutions – Related Work

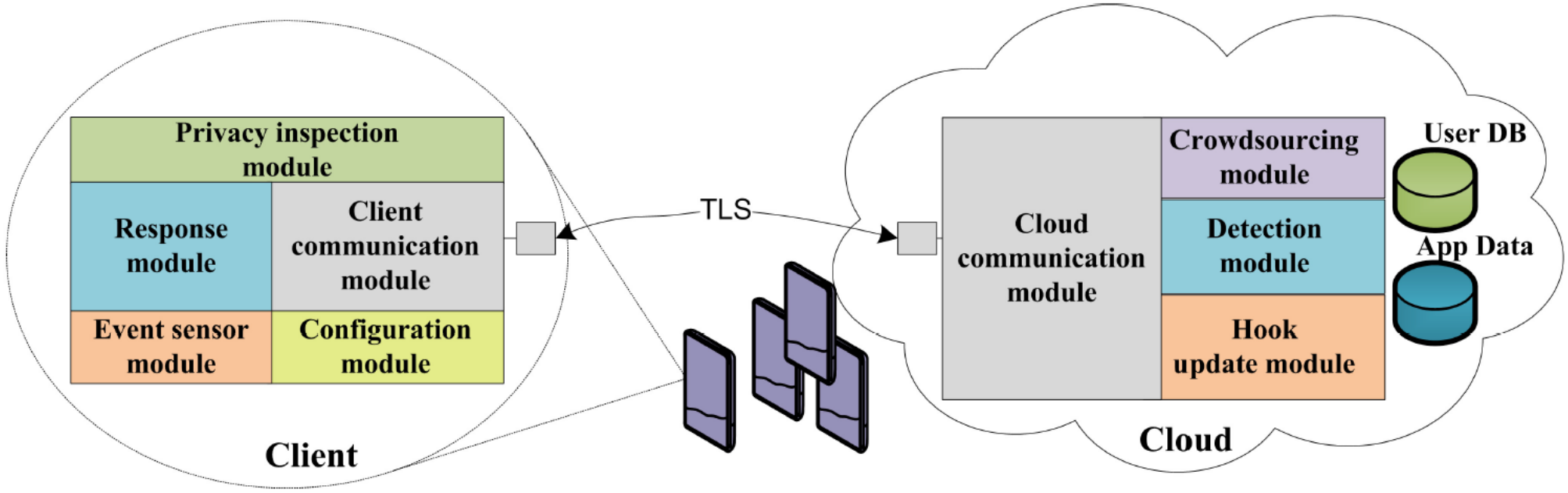
- Malware Detection
  - Based on Linux kernel system calls.
  - Virtual exact replicas of the smartphone in the cloud for applying sophisticated detection techniques.
  - High communication overhead, not "really real-time protection"
- Detection of privacy leaks
  - Many static analysis tools aim to analyze mobile apps behavior.
  - Personal information tracking.
  - Lately, attempts have been made to somehow quantify privacy invasion or exposure level.

Our Contribution

- A host and cloud-based synergistic mechanism for preserving mobile users' privacy.
- The formation and diffusion of a common knowledge about virtually any mobile app.
  - Sharing apps' behavioral profile among the participating users.
- A continually revised product of a collaborative effort fusing the user's preferences for sharing the actual app behavior in human understandable form.
- The solution relies on cloud discharging smartphones from resource intensive tasks.

Our Proposal

- Three main services are provided:
  - privacy-flow tracking
  - crowdsourcing
  - detection and reaction against privacy violations

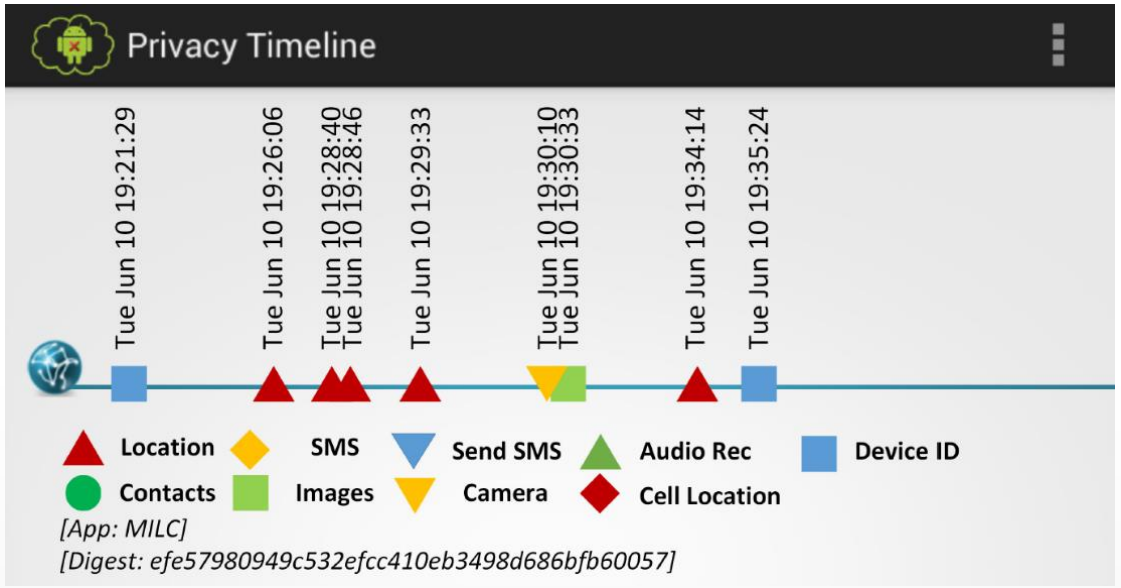


The Client

- **Event sensor module:**
  - Hooking methods' invocation and objects' creation – Cydia Substrate.
  - Privacy-sensitive assets and their associated methods has to be decided beforehand.
  - The sensor stores an Event Vector for every incident.
  - EV={ appID, hookID, hookedMethodID, hookMode, rFlag, timestamp }
- **Configuration module:**
  - GUI where users can personalize their resource accessing preferences per app and per resource.
  - Mode 1: Always Allowed (default permit)
  - Mode 2: Ask me first
  - Mode 3: Always Blocked (default deny)
  - Parameterization recommendation derived from cloud.

The Client

- **Privacy inspection module:**
  - Provides a privacy flow tracking service able to inform users about their privacy exposure level.
  - Responsible for visualizing events happening on privacy-sensitive assets.



- **Client communication module:**
  - Transmits Enhanced EVs (EEVs)
  - EEV= {deviceID, appID, appName, apkDigest, hookID, hookedMethodID, hookMode, rFlag, timestamp}

The Client

- **Response module:**
  - Based on users' privacy preferences this module acts upon any privacy sensitive method invocation.
  - Constantly tracks the user response per app hook for setting the rFlag value.

The Cloud

- **Cloud communication module:**
  - It receives data related to the behavior of the various apps running on the clients.
  - Disseminates collective data about the apps.
  - Upon a new app install happening on the client it sends collective behavior data about the app.
  - Multicasts an update for one or several apps to a group of clients.

The Cloud

- **Crowdsourcing module:**
  - Based on the privacy configuration among users provides configuration suggestions.
  - Statistical calculations
    - Mean value of trust among users.
    - Standard Deviation over the mean trust value – conceptualize the degree of concordance among users.
- Example:
  - 10 hooks in use for an app.
  - User configuration: 3 in Mode 3, 5 in Mode 2, and the rest in Mode 1
  - 3*3+5*2+2=21, which corresponds to a parentage of 70% (user's trust is 30%)
  - 5 users with 30, 80, 90, 95 and 85% →Mean: 76% , St. Deviation: 26.3

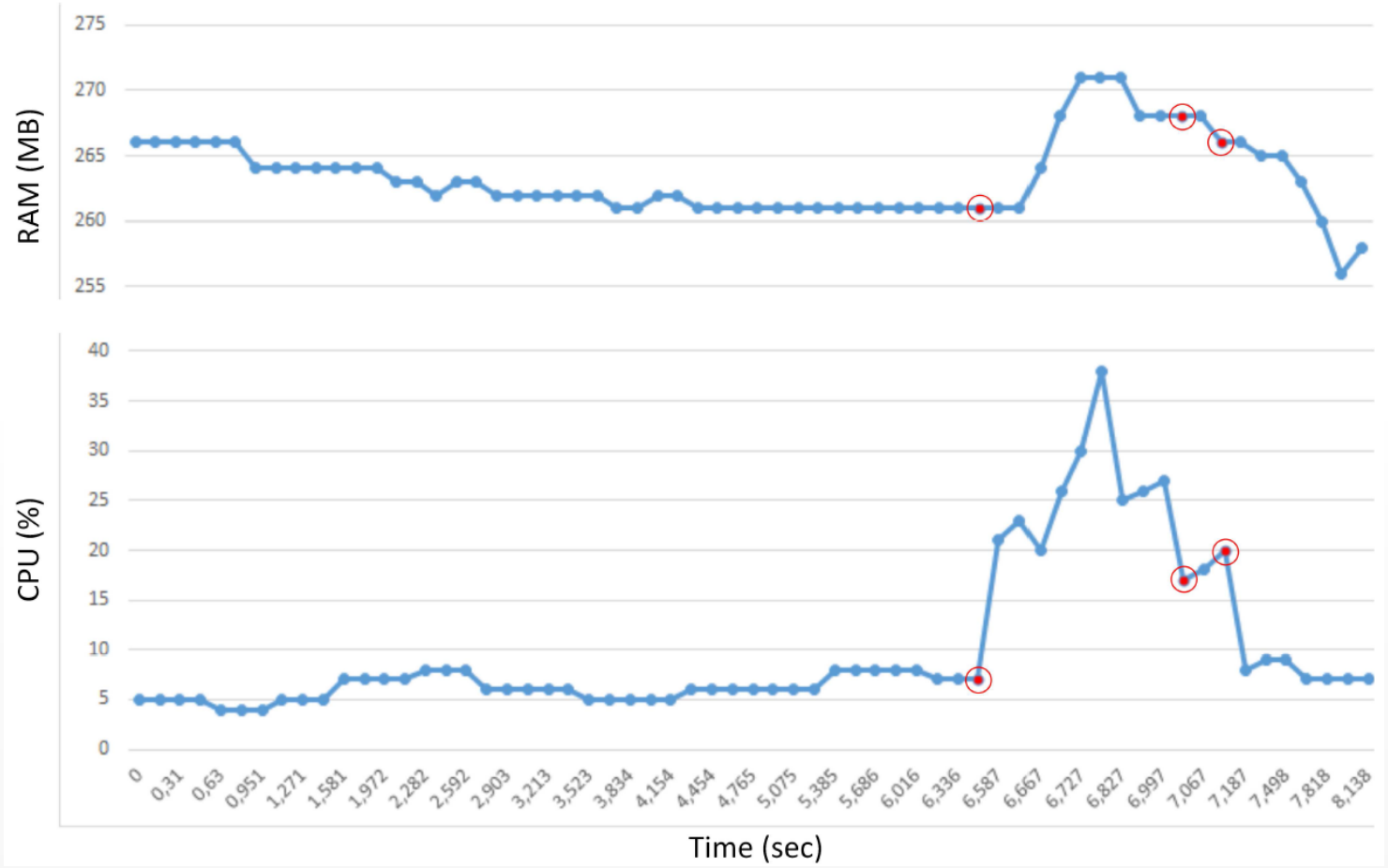
The Cloud

- **Detection module:**
  - Execution traces contributes toward the design of a behavioral-based detection mechanism for exposing suspicious and/or sudden changes in the behavior of an app.
  - The current module is destined to deliver such a service in a future work.
- **Hook update module:**
  - Updates regarding the hook list on the client side.
  - Straightforward functionality if we consider the open nature of the Google's Android platform.
  - Manufacturers design their own custom APIs to meet their specific needs.
  - Frequent OS updates.

Evaluation

- Preliminary results reveal the efficiency of our mechanism.
- Performance is directly associated with Cydia Substrate tool.
- Sony Xperia L device which incorporates a dual core 1 GHz CPU and 1 GB of RAM.
- The maximum CPU consumption during tests is 38% while the memory increased by 10 MB.
- The time penalty for detecting, suppressing and logging a sensitive resource access is 0.6 secs.

Evaluation


RAM (MB)
275
270
265
260
255
CPU (%)
40
35
30
25
20
15
10
5
0
0
0,31
0,63
0,951
1,271
1,581
1,972
2,282
2,592
2,903
3,213
3,523
3,834
4,154
4,454
4,765
5,075
5,385
5,686
6,016
6,336
6,587
6,667
6,727
6,827
6,997
7,067
7,187
7,498
7,818
8,138
Time (sec)

Conclusion

- Sharing of a common knowledge per app in an understandable form.
- The decisions about an app stem from both the user's personal preferences and the crowdsourced knowledge in the cloud.
- The device is relieved from resource demanding tasks.
- Distribute the task and responsibility of controlling app privacy to all the community, rather than dealing with specific issues in an isolated manner.

Thank you!