AppProfiler: A Flexible Method of Exposing Privacy-Related Behavior in Android Applications to End Users

Sanae Rosen (sanae@umich.edu), Zhiyun Qian (zhiyunq@nec-labs.com), Z. Morley Mao (zmao@umich.edu)

1University of Michigan, Ann Arbor; 2NEC Labs

Introduction

Problem: Hard to make informed decisions about what applications to install on mobile devices such as Android. Permissions insufficient for this purpose.

Solution: Automatically extract profiles of application behavior offline. Make profiles available to users through a mobile application.

How?: Use a knowledge base mapping API call patterns to meaningful privacy-relevant behaviors.

Some benefits:
- Can be implemented now, without changes to apps or Android
- Flexible to changes in API, new discoveries, different audiences
- Allows large-scale in-depth investigation of app behavior trends

System Overview

Apps...

Download and Decompile
Extract API calls
Apply knowledge base to identify pertinent behaviors
Large-Scale Analysis
Detailed Profiles

Download and Unzip
Extract APIS
Apply Knowledge Base
Identify Perinent Behaviors

User Feedback

Permissions are not specific enough:
Examples:
- Use of cookies flagged as objectionable 23% of time vs 6% of time for internet use in general
- Photo without preview flagged 66% of time vs 26% for camera preview only

Behavior not covered by permissions can be objectionable:
E.g. use of hardware sensors, checking the applications installed is worse than network usage.

Users care about non-malicious but privacy-intrusive behavior:
- Top 10 behaviors disliked by users involve location data, cookies, unique device IDs.
- Malware-specific behavior, especially that involving SMS messages, were not of concern to users.

Step 1: Create Knowledge Base

List of potential API calls

Identify high-priority, privacy-relevant classes

Existing research

Refine mappings with domain-specific knowledge

Frequency analysis of classes and methods

Mapping of API call patterns to behavior labels

Example knowledge base entry:

Category: Location - Type
Subcategory: Regional data – State

FunctionCall call:
call.function.enclosingClass.name
startsWith
"android.location.Address"
and call.function.name == "getAdminArea"

FunctionCall call:
call.function.enclosingClass.name
startsWith
"android.location.Address"
and call.function.name == "getSubAdminArea"

Website:
approfiles.eecs.umich.edu

Step 2: Apply to Apps

Process Apps...

Find rule matches in decompiled app

Identify context of behavior (user input, background service)

Identify behaviors from multiple rules (e.g. photo with no preview)

Identify which rules belong to ad libraries

Convert data into user-readable profiles

Simplify and focus on key behaviors

...And Make Available to Users

Technical Profile Example
- Use GPS or network
- latitude/longitude (broadcast receiver)
- Updates every 1/2 or less (activity) (jumptap library)
- Proximity to location (activity, on click)

User Profile Example
- Gather precise location data (e.g. GPS)
- Uses more of your phone's resources than recommended to gather location data
- Concerned with your proximity to a given location

Large-Scale App Analysis

Used profiles to perform a survey of app behavior. Key findings:
- Substantial differences between popular apps and market as a whole
- Behavior of third-party libraries differs greatly from application-specific behavior
- Many popular third-party libraries pose privacy concerns; several others are unintrusive
- Some permission-heavy popular applications, (e.g. Angry Birds), share a lot of personal data with ad libraries. Others, (e.g. Facebook), collect private data largely in direct response to user actions.