Ubiquitous and Mobile Computing
CS 528: Information Leakage through Mobile Analytics Services

Punit Dharani
Evin Ugur

Computer Science Dept.
Worcester Polytechnic Institute (WPI)
Overview

- Introduction – EU
- Related Work – PD
- Extracting User Profiles – EU
  - Methodology
  - Validation
- Influencing Advertisements - PD
  - Methodology
  - Validation
- Implications - PD
- Countermeasures – EU
- References
Introduction

What?
- In-App Ads are a popular revenue model for app developers
- Profiles for Ad Services contain sensitive information, and can be extracted
- With these profiles compromised, ads served can be influenced.

Why?
- Privacy Concerns! - $$$
Related Work

- **Don't kill my ads!: balancing privacy in an ad-supported mobile application market**
  - feedback control loop for AD privacy adjustment

- **MAdFraud: Investigating Ad Fraud in Android Application**
  - Methods for identifying ad fraud – we will soon present a way to create ad fraud
Introduction - Background

The Mobile Ad Ecosystem

Methodology

Two Phases:
1) Extraction of User Profiles
2.) Influencing Ads Served
Methodology: 1.) Extraction of User Profiles

- User Profile – a set of info collected or inferred by the ad service
  - Basic: Age, Gender, Language, Geography
  - Creepy: Singles, New-Moms, High Net-Worth
- Extraction is performed by impersonating the user, and ultimately performing actions on their behalf
  - Google – identified by Android ID, triggered from AD Settings
  - Flurry must cause communication with bespoke app
Methodology: 1.) Extraction of User Profiles (Continued)

- Monitor the network for device IDs
  - On a public hotspot? Throw up a net and capture 1000s of IDs
  - Private Network? Capture your friend, coworker, etc.
- Modify values of identified parameters on a rooted Android Device & You've Spoofed your Target
Methodology: 1.) Extraction of User Profiles (Continued)
Validation: 1.) Extraction of User Profiles

- Experiment with 44 Users – aim is to show they can be spoofed
  - Instantiate a new usage report from ad service on real device and from a spoof with the same app ID
  - Report served has identical device IDs despite being run on different devices
Methodology: 2.) Influencing Ads Served

- Impersonating target devices using Spoofed user profiles

Profile Training – training the user profiles by running apps from a targeting category i.e Business apps

Perturb a profile – running app from different categories for significantly longer periods to set a new dominant category
Methodology: 2.) Influencing Ads Served (Continued)

Ad collection: in-app ads delivered via HTTP
tcpdump on Android to monitor ad traffic
Captured traffic pulled from device every 10 minutes
Validation: 2.) Influencing Ads Served

- Jaccard Index between set of unique ads received by all profiles

![Figure 6: Unique ads similarity before and after profile perturbation. (H - high, M - moderate and L - low)]
Implications

- Exposure of personal information
- Malicious attacks increasingly sophisticated
- Industry awareness (manufacturers, OS, advertisers, etc.)
- Theoretical compromise of entire monetization model
Countermeasures

- Google hashes Device ID
  - Not strong enough since it can still be sent by other libraries in plain text and then trivially mapped to the hash
- Implement user ID & advertising ID
  - Lets users reset their profiles – akin to clearing cookies in a browser
- Utilize SSL – Conflict of Interests with Ad
- Public Key Signing Model with Ad Network
  - Uses certificates; Powerful, but not practical – industry wide effort to implement
Countermeasures (Continued)

- Using SSL Prevents Easy Interception – but adds Bandwidth
- Increases ad load time – conflict of interest
- Eats into data plans on the aggregate of those with limited data

<table>
<thead>
<tr>
<th>protocol</th>
<th>onStartSession</th>
<th>getAds</th>
<th>total/hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>latency</td>
<td>bandwidth</td>
<td>latency</td>
</tr>
<tr>
<td>HTTP</td>
<td>160±1 ms</td>
<td>422 B</td>
<td>160±1 ms</td>
</tr>
<tr>
<td>HTTPS</td>
<td>800±5 ms</td>
<td>3288 B</td>
<td>800±5 ms</td>
</tr>
</tbody>
</table>
References