Comparing Mobile Privacy Protection through Cross-Platform Applications

“iOS vs. Android”

Jin Han*, Qiang Yan†, Jianying Zhou*, Debin Gao†, Robert Deng†

*Institute for Infocomm Research, Singapore
†Singapore Management University
Comments from Media

Android 🍃

Why Android App Security Is Better Than for the iPhone

Android, iPhone security different but matched

iOS 🍎

Android much less secure than iPhone
Comparison via Cross-platform Apps

- Our solution – comparing the **cross-platform apps** running on Android and iOS:
  - Designed to provide the **same core functionalities**
  - Released by the same developer/company
  - Similar user interfaces and visible features
What to compare -- Usage of SS-APIs

- **Security-Sensitive APIs (SS-APIs)**
  - Provide access to user sensitive data
    - • Contacts, Calendar, SMS, ...
  - Provide access to hardware features
    - • Bluetooth, Camera, Audio Recorder, Vibration ...
  - Multiple **SS-APIs** → A type of SS-APIs ≈ A privilege
    - • Borrow/refine the permission classification from Android.

- SS-API usage ≈ Privilege usage
### Privileges supported by both platforms

<table>
<thead>
<tr>
<th>Privilege (SS-API Type)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS сетуATION</td>
<td></td>
</tr>
<tr>
<td>ACCESS сетуATION_INFO</td>
<td></td>
</tr>
<tr>
<td>BATTERY STATS</td>
<td></td>
</tr>
<tr>
<td>BLUETOOTH</td>
<td></td>
</tr>
<tr>
<td>BLUETOOTH_ADMIN</td>
<td></td>
</tr>
<tr>
<td>CALL_PHONE</td>
<td></td>
</tr>
<tr>
<td>CAMERA</td>
<td></td>
</tr>
<tr>
<td>CHANGE сетуITION сетуCAST сетуATE</td>
<td></td>
</tr>
<tr>
<td>FLASHLIGHT</td>
<td></td>
</tr>
<tr>
<td>INTERNET</td>
<td></td>
</tr>
<tr>
<td>READ сетуATION сетуAR</td>
<td></td>
</tr>
<tr>
<td>READ сетуATION сетуNTAC TS</td>
<td></td>
</tr>
<tr>
<td>READ сетуATION сетуDEVICr_ID</td>
<td></td>
</tr>
<tr>
<td>RECORD сетуATION AUDIO</td>
<td></td>
</tr>
</tbody>
</table>

### SS-APIs on Android:
- `android.location.LocationManager.addGpsStatusListener()`
- `android.location.LocationManager.getProvider()`
- `android.telephony.TelephonyManager.getCellLocation()`
- `android.telephony.TelephonyManager.getNeighboringCellInfo()`
- `android.webkit.GeolocationService.setEnableGps()`
- ...

### SS-APIs on iOS:
- `[CLLocationManager startUpdatingLocation]`
- `[CLLocationManager startMonitoringSignificantLocationChanges]`
- `[CLLocationManagerDelegate locationManager:didUpdateToLocation:fromLocation:]`
- `MKUserLocation.location`
- ...

Privileges supported by both platforms:

- `ACCESS_COARSE_LOCATION`
- `ACCESS_FINE_LOCATION`
**Methodology Overview**

1. **Identify and sample Cross-Platform Apps**
   - 1a. *Web crawlers* for Google Play (300,000) and iTunes Store (400,000)
   - 1b. *App matcher* based on information retrieval techniques

2. **Establish SS-API Mappings between Android and iOS**
   - 2a. Identify iOS SS-APIs
   - 2b. Classify iOS & Android SS-APIs

3. **Perform** Static Analysis on binary files of Android and iOS apps
   - 3a. Static Analysis Tool for iOS Objective-C Executable
   - 3b. Static Analysis Tool for Android Dalvik Bytecode
   - 3c. SS-API analyzer for SS-API separation and comparisons
Results at First Glance

For **1300** pairs of popular free cross-platform apps:

- Certain privileges (INTERNET, BLUETOOTH) are required almost equally.
- Many other privileges are required very **differently**.
- **948 (73%)** of iOS apps access additional privileges compared to its Android version.
Privilege Usage Difference

<table>
<thead>
<tr>
<th>Privilege</th>
<th># of Android Apps</th>
<th># of iOS Apps</th>
<th>Only on iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ_DEVICE_ID</td>
<td>510</td>
<td>925</td>
<td>469</td>
</tr>
<tr>
<td>CAMERA</td>
<td>172</td>
<td>601</td>
<td>435</td>
</tr>
<tr>
<td>VIBRATE</td>
<td>374</td>
<td>522</td>
<td>290</td>
</tr>
<tr>
<td>ACCESS_NETWORK_INFO</td>
<td>885</td>
<td>1065</td>
<td>269</td>
</tr>
<tr>
<td>READ_CONTACTS</td>
<td>151</td>
<td>388</td>
<td>256</td>
</tr>
<tr>
<td>SEND_SMS</td>
<td>29</td>
<td>264</td>
<td>248</td>
</tr>
<tr>
<td>READCALENDAR</td>
<td>35</td>
<td>174</td>
<td>141</td>
</tr>
</tbody>
</table>

- iOS apps usually access more privileges than Android apps, which are often associated with accessing sensitive resources such as device ID, camera, and users’ contacts.
Case #1: Angry Birds

- The almighty game by Rovio
  - requires **READ_CONTACTS** on iOS

- API call `ABAddressBookGetPersonWithRecordID` observed in the code section of `CCPrivateSession`. `getArrayOfAddressBook EmailAddressesNames AndContactIDs`

  **Still exist** until version 2.1.0
  (released in **March 2012**)

  **Removed** on version 2.2.0
  (released in **August 2012**)
Case #2: Words With Friends

• A famous game app by Zynga
  – iOS version requires 13 privileges.
  – Android version only requires 6.

• The additional privileges on iOS:
  – **BATTERY_STATS**
    API call UIDevice.setBatteryMonitoringEnabled in the code region of MMMManager.handshakeURL [Millennial Media]
  – **CALL_PHONE**
    UIApplication.openURL with “tel:” parameter in IMAdView.placeCallTo and other locations
  – **CAMERA**
    UIImagePickerController.setSourceType is observed in MobclixRichMediaWebAdView.takePhotoAndReturnToWebview
• Privilege Usage of Third-Party Libraries
  – We identified commonly used third-party libraries on both Android (79 libraries) and iOS (72 libraries).

<table>
<thead>
<tr>
<th>Library Name</th>
<th>Android App Ratio</th>
<th>iOS App Ratio</th>
<th>SS-API Types on Android</th>
<th>SS-API Types on iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Ads</td>
<td>21.7 %</td>
<td>15.9 %</td>
<td>ANI, INT</td>
<td>ANI, INT, RDI, SMS, VIB, WAK</td>
</tr>
<tr>
<td>Flurry</td>
<td>19.1 %</td>
<td>19.9 %</td>
<td>LOC, INT</td>
<td>LOC, INT, RDI</td>
</tr>
<tr>
<td>Millennial Media</td>
<td>7.3 %</td>
<td>9.3 %</td>
<td>ANI, INT, RDI</td>
<td>LOC, ANI, CAM, INT, CON, RDI, VIB</td>
</tr>
<tr>
<td>AdWhirl</td>
<td>3.8 %</td>
<td>6.9 %</td>
<td>LOC, INT</td>
<td>LOC, ANI, INT, RDI</td>
</tr>
<tr>
<td>Mobclix</td>
<td>3.2 %</td>
<td>3.7 %</td>
<td>LOC, ANI, INT, RDI</td>
<td>LOC, ANI, BAT, CAM, FLA, INT, CAL, CON, RDI, SMS, VIB</td>
</tr>
</tbody>
</table>
Investigation #2: Apps’ Own Code

- Corresponding security sensitive APIs may also be accessed by the App’s own code.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Exclusively caused by Lib</th>
<th>Exclusively caused by App</th>
<th>Caused by both Lib &amp; App</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ_DEVICE_ID</td>
<td>36%</td>
<td>40%</td>
<td>24%</td>
</tr>
<tr>
<td>CAMERA</td>
<td>27%</td>
<td>62%</td>
<td>11%</td>
</tr>
<tr>
<td>VIBRATE</td>
<td>54%</td>
<td>38%</td>
<td>8%</td>
</tr>
<tr>
<td>ACCESS_NETWORK_INFO</td>
<td>4%</td>
<td>86%</td>
<td>10%</td>
</tr>
<tr>
<td>READ_CONTACTS</td>
<td>25%</td>
<td>48%</td>
<td>27%</td>
</tr>
<tr>
<td>SEND_SMS</td>
<td>32%</td>
<td>51%</td>
<td>17%</td>
</tr>
<tr>
<td>READ_CALENDAR</td>
<td>33%</td>
<td>65%</td>
<td>2%</td>
</tr>
</tbody>
</table>

* This table shows the usage pattern for those extra privileges only used in iOS apps.
Possible Explanation #1

- **Functional difference**
  - `ACCESS_NETWORK_INFO`
    - Caused by the implementation difference on the Reachability test by analyzing several open-source apps.
  - `CAMERA`
    - OpenFeint library on iOS and Android:
      - Use `CAMERA` only on iOS, for setting profile photos.
      - Every game with OpenFeint enabled would require `CAMERA` privilege.
Possible Explanation #2

- **Intentional avoidance**
  - *WordPress* app on Android obtains **UUID** differently compared to its iOS version
  - The programmers intentionally avoid triggering **READ_PHONE_STATE** on Android.
  - Confirmed by consulting WordPress developers:

  “... *because it doesn’t require that permission which reads quite poorly as ‘read phone state and identity’...*”
The Implication

Android apps show its privilege list before installation

Most users may not read/understand this privilege list...

However, as long as the app developers realize someone may read it, they become more cautious in requesting privileges.
Evolution on iOS

- The original comparison was performed on iOS 5.0 and Android 4.0
  - On iOS 5, only two privileges are shown to user:
    • access location info & send push notifications
  - Since iOS 6, more privileges can be controlled:
    • access to contacts, calendar, photos and reminders.

- Such changes have impacts on privilege usage:
  - 18% iOS apps originally require READ_CONTACTS have removed this privilege in their new versions.
  - 16% removed for READCALENDAR privilege.
Evolution on iOS

• The original comparison was performed on iOS 5.0 and Android 4.0
  – On **iOS 5**, only two privileges are shown to user:
    • access location info & send push notifications
  – Since **iOS 6**, more privileges can be controlled:
    • access to contacts, calendar, photos and reminders.

• Such changes have impacts on privilege usage:
  – API call `ABAddressBookGetPersonWithRecordID` observed in AngryBirds
  – Still exist until version 2.1.0 (Mar 2012)
  – **Removed** from version 2.2.0 (Aug 2012)
Conclusion

• This work is the **first attempt** to establish a baseline on **systematic comparison between** Android and iOS, which shows how the platform difference affects the behavior of **cross-platform apps**.

• Our results show
  – iOS apps turn to access more **Security-Sensitive APIs**, which are related to sensitive resources such as device ID, contacts and calendar.
  – Caused by both **third-party libraries** and apps’ own code.
  – A **strong correlation** exists between the usage difference of **privileges** and the **availability** of **privilege-list** mechanism on Android and iOS.
iOS vs. Android?

<table>
<thead>
<tr>
<th>Security Feature</th>
<th>Android</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permission Notification</td>
<td>Yes</td>
<td>Little</td>
</tr>
<tr>
<td>Approval/Vetting Process</td>
<td>Partial</td>
<td>Yes</td>
</tr>
<tr>
<td>Binary Encryption</td>
<td>Since v4.1</td>
<td>Yes</td>
</tr>
</tbody>
</table>

• Android – open source platform
• iOS – closed source platform

• How to compare?
## Android permission classification

<table>
<thead>
<tr>
<th>Group of Privileges</th>
<th># of Privileges</th>
<th>SS-API Type Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not actually exist in Android</td>
<td>7</td>
<td>SET_PREFERRED_APPLICATIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BRICK</td>
</tr>
<tr>
<td>Reserved for System or OEMs</td>
<td>42</td>
<td>DELETE_CACHE_FILES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WRITE_SECURE_SETTINGS</td>
</tr>
<tr>
<td>Not supported by iOS</td>
<td>46</td>
<td>CHANGE_NETWORK_STATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MODIFY_AUDIO_SETTINGS</td>
</tr>
<tr>
<td>Supported by both Android and iOS</td>
<td>20</td>
<td>BLUETOOTH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>READ_CONTACTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECORD_AUDIO</td>
</tr>
<tr>
<td>Total</td>
<td>115</td>
<td></td>
</tr>
</tbody>
</table>
Android Permission Notification

A music player

Gmail

Facebook

This application has access to the following:

- **Network communication**
  - Full Internet access

- **Your personal information**
  - Read contact data, write contact data

- **Network communication**
  - Full Internet access

- **Your accounts**
  - Google mail, manage the accounts list, use the authentication credentials of an account

- **Storage**
  - Modify/delete USB storage contents

- **System tools**
  - Prevent phone from sleeping, write sync settings

- **Your accounts**
  - Act as an account authenticator, manage the accounts list

Show all
- Class inheritance
- Java reflection resolving
- Content provider
- Intents
iOS App Static Analysis Tool

- iOS static analysis tool:
  - App decryption/cracking
  - Method boundaries marking
  - `objc_msgSend` resolving