Anatomization and Protection of Mobile Apps’ Location Privacy Threats

Kassem Fawaz, Huan Feng, and Kang G. Shin
Computer Science & Engineering
The University of Michigan
Apps’ Location Privacy

What can we do about it?
Apps’ Location Privacy

• More than a decade of research
• Lots of location privacy protection mechanisms
  – E.g., LP-Guardian, MockDroid, Caché, Koi, etc.
• Few implementations/deployments:
  – Some have not been implemented with real-world apps
  – Others very difficult to deploy (require custom ROMS, etc.)
• We are left with what OSes provide us
Mobile OS Location Controls

- Install time permissions
- Per-app location switch
- Background location access

http://arstechnica.com/gadgets/2015/05/android-m-dev-preview-launches-permission-controls-fingerprint-api-and-more/
Apps’ Location Privacy

• Are OS-based controls effective in protecting the user’s location privacy?

• If not, how can they be improved, in user-level, without modifying any app or the OS?
This Talk

Measurement
- Collect app-tagged location traces
- Study location access and usage patterns of different apps

Anatomy
- Quantify location privacy threat for apps
- Analyze effectiveness of OS controls

LP-Doctor
- Leverage OS controls to balance utility and privacy
- Easy to install and use

Measurement and LP-Doctor are based on Android, but conclusions apply to other platforms
Measurement
Measurement

App runs in:

- Foreground
- Cached
- Background

Location-access pattern = F(app) → offline analysis

App-usage pattern = F(user) → user-level collection

Location granularity & frequency

App-tagged location traces

- <app, x, y, e, t_1>
- <app, x, y, e, t_2>
- <app, x, y, e, t_3>

App runs in:

- Foreground
- Cached
- Background

Location granularity & frequency

App-usage pattern = F(user) → user-level collection
# Location-Access Patterns

<table>
<thead>
<tr>
<th>Location permissions</th>
<th># apps</th>
<th>Foreground (%)</th>
<th>Cached (%)</th>
<th>Background (%)</th>
<th>None (%)</th>
<th>Coarse Granularity enough (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>210</td>
<td>71</td>
<td>6</td>
<td>1</td>
<td>22</td>
<td>100</td>
</tr>
<tr>
<td>Fine</td>
<td>955</td>
<td>74</td>
<td>14</td>
<td>4</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td>All</td>
<td>1165</td>
<td>74</td>
<td>12</td>
<td>3</td>
<td>14</td>
<td>66</td>
</tr>
</tbody>
</table>

- Apps abuse the location permissions
- Minority of the apps continuously access location
- Focus on foreground location access
App-Usage Patterns

- App Usage Patterns Analysis
  - List of app sessions:

    com.whatsapp,1395247179636,America/New_York,75,placeID:1,placeID:1

- Three datasets:

<table>
<thead>
<tr>
<th></th>
<th>RTCL</th>
<th>PhoneLab</th>
<th>LiveLab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participants</td>
<td>25</td>
<td>95</td>
<td>30</td>
</tr>
<tr>
<td>Period</td>
<td>1 week - 10 months</td>
<td>1 week - 4 months</td>
<td>1 year</td>
</tr>
<tr>
<td>Location</td>
<td>Ann Arbor, MI</td>
<td>Buffalo, NY</td>
<td>Houston, TX</td>
</tr>
</tbody>
</table>
Location-Access Model

- App session maps to a place the user visited
  - 98% of app sessions started and ended at the same place

- Model app as a histogram
  - Map place to number of visits
  - Sample of user’s mobility
Anatomy

Location Privacy Metrics

Capture and quantify the user’s original mobility pattern with the app’s histogram

- Probability of user visiting place
- # of visits app observed user at place
Location Privacy Metrics

• Four privacy metrics:
  
  – $Pol_{total}$: Portion of user’s identified significant places
    • The closer to 1, the more places are identified
  – $Pol_{sens}$: Portion of identified rarely visited places
    • Represent “deviant” behaviors which are more sensitive
  – $Prof_{cont}$: KL divergence between histogram and mobility pattern
    • Distance between app’s observation and user’s mobility profile
  – $Prof_{bin}$: $\chi^2$ test with significance level of 0.05
    • Whether the histogram (sample) is a good fit of the mobility pattern
# Threat Anatomy

<table>
<thead>
<tr>
<th>Metric</th>
<th>Insight</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>For each user: 20% of apps identified most PoIs</td>
</tr>
<tr>
<td>PoIs</td>
<td>Most apps identified users' top two PoIs</td>
</tr>
<tr>
<td>sens</td>
<td>For each user: 20% of apps identified most sensitive PoIs</td>
</tr>
<tr>
<td>A&amp;A</td>
<td>A&amp;A libraries identified more of the sensitive PoIs</td>
</tr>
<tr>
<td>prof</td>
<td>No correlation between app usage frequency and privacy threats</td>
</tr>
<tr>
<td>cont</td>
<td>A&amp;A posing significantly more threats</td>
</tr>
<tr>
<td></td>
<td>Value mostly larger than 0</td>
</tr>
<tr>
<td></td>
<td>Histograms are not uniform samples of user’s mobility</td>
</tr>
</tbody>
</table>
## Threat Anatomy

<table>
<thead>
<tr>
<th>Metric</th>
<th>Insight</th>
</tr>
</thead>
</table>
| $Pol_{total}$ | • For each user: 20% of apps, each, identified most Pols  
|           | • Most apps identified users’ top two Pols                               |
## Threat Anatomy

<table>
<thead>
<tr>
<th>Metric</th>
<th>Insight</th>
</tr>
</thead>
</table>
| $Pol_{total}$ | • For each user: 20% of apps, each, identified most PoIs  
• Most apps identified users’ top two PoIs |
| $Pol_{sens}$ | • For each user: 20% of apps identified most sensitive PoIs  
• A&A libraries identified more of the sensitive PoIs |
## Threat Anatomy

<table>
<thead>
<tr>
<th>Metric</th>
<th>Insight</th>
</tr>
</thead>
</table>
| $Pol_{total}$ | • For each user: 20% of apps, each, identified most PoIs  
• Most apps identified users’ top two PoIs                                                                                                                                                          |
| $Pol_{sens}$ | • For each user: 20% of apps identified most sensitive PoIs  
• A&A libraries identified more of the sensitive PoIs                                                                                                                                                 |
| $Prof_{cont}$ | • No correlation between app usage frequency and privacy threats  
• A&A posing significantly more threats  
• Value mostly larger than 0  
  • histograms are not uniform samples of user’s mobility                                                                                                                                           |
Are OS-Controls Effective?

• Android location permissions
  – Fail as a notification and choice mechanism
    • App-usage behavior is independent from app’s location permissions

• Background location blocking
  – Limits tracking (less than 10 minutes a day) but not profiling

• Per-app permissions
  – Need to block location access of 10% of apps to neutralize each A&A library
  – Take-it or leave-it situation
    • Users can’t control when and where an app can access location
LP-Doctor
LP-Doctor: WHY?

• Per-session location access control is needed
  – Existing OSes don’t provide an easy interface
  – User can’t decide on threat level, especially for A&A libraries

• Our Solution: LP-Doctor – a user-level tool that:
  – Leverages and complements existing OS-controls
  – Is easy to install and use
Threat Model

• What’s in?
  – Honest-but-curious adversaries
    • Service providers or Advertisement and Analytics (A&A) agencies
    • Access location only through location APIs
    • Foreground location access

• What’s out?
  – Background location access
    • Some platforms are providing controls like iOS
  – Operating systems and cellular operators
    • Users have no choice but to trust them
  – Security issues
LP-Doctor

What happens when the user installs an app?

Apply protection before each session

Level decides:
• Amount of noise added
• Sensitivity level for $\text{prof}_{\text{bin}}$

New App Detected

You just installed Uber. This app can access your accurate location information.

What behavior do you want applied?
- Allow Location Access
- Block Location Access Completely

Protect Me when dangerous
Protection Level: low

Done
LP-Doctor

What happens when the user runs an app?

1. Instrument the CyanogenMod app launcher
2. Pause app execution
3. Divert execution to LP-Doctor
What happens when the user runs an app?

**LP-Doctor**

- **Intercept app launch event**
- **Fetch policy**
- **Threat analyzer**
- **Instruct app to launch**
- **Display notification**
- **Anonymization actuation**

**Compute** $\text{prof}_{\text{cont}}$ for before ($m_{\text{bef}}$) and after histogram ($m_{\text{aft}}$)

$m_{\text{bef}} > m_{\text{aft}}$

- **Yes**
  - Compute $\text{prof}_{\text{bin}}$
  - **Yes**
    - Protect location
  - **No**
    - Release location

- **No**
  - Release location

$\text{prof}_{\text{bin}} = 1$?
LP-Doctor

What happens when the user runs an app?

1. **Compute** obfuscated location
   - Add 2-D Laplacian noise that achieves indistinguishability
2. **Engage** Android’s mock location provider
3. **Apply** same obfuscated location for future sessions of the same app from the place
   - Reduces information leaks
LP-Doctor

What happens when the user runs an app?

- Intercept app launch event
- Fetch policy
- Threat analyzer
- Instruct app to launch
- Display notification
- Anonymization actuation

Yelp
LP-Doctor is adding noise to location.

Remove
Reduce

Remove noise
Reduce noise → user adjusts privacy – QoS trade-off
LP-Doctor

What happens when the user runs an app?

Intercept app launch event → Fetch policy → Threat analyzer → Anonymization actuation

Instruct app to launch → Display notification

1. Resume app launch
2. Delay less than 25 ms
3. End of app session:
   - Update app histogram
   - Disengage mock location provider
User Study

- Recruited 227 participants from Amazon Mechanical Turk
- Used LP-Doctor with Yelp ($N = 120$) and Facebook ($N = 122$)

Enable mock locations on Android
Reinstall Facebook or Yelp
Search for restaurant using Yelp
Check-in from a nearby place using Facebook

One time effort
Installation menu
Notifications & Impact on QoS
User Study

Installation Menu

- Use again
- Useful
- Control
- Informative
- Usable

Appears only once for each 5 installed apps

Notification

- Use again
- Yelp
- Facebook
- Usable

Appears for 12% of sessions on average
User Study

LP-Doctor adds noise automatically for 12% of the sessions on average of each app

Yelp
- Coarse location: Runs normally
- No change in user experience: Runs normally
- Relevant results: Runs normally
- Runs normally: Runs normally

Facebook
- Coarse location: Runs normally
- No change in user experience: Runs normally
- Relevant places: Runs normally
- Runs normally: Runs normally
User Study

Post study questions

**Yelp**
- Comfortable with fake location?
  - I don't know: 6%
  - No: 13%
  - Yes: 81%

**Facebook**
- Comfortable with fake location?
  - I don't know: 11%
  - No: 10%
  - Yes: 79%

**LP-Doctor**
- Install LP-Doctor or similar tool?
  - I don't know: 11%
  - No: 11%
  - Yes: 78%
Conclusion

• Location privacy is a serious problem
  – Existing research proposals are impractical
• Are OS-controls enough?
  – Measured threats from > 400 apps as used by > 100 users
  – Found that OS-controls are ineffective and inefficient
• We propose LP-Doctor that is practical and effective
• Future work:
  – Test LP-Doctor in the wild to study deployment challenges
Thank You!

Questions?

Kassem Fawaz
kmfawaz@umich.edu
github.com/kmfawaz/LP-Doctor