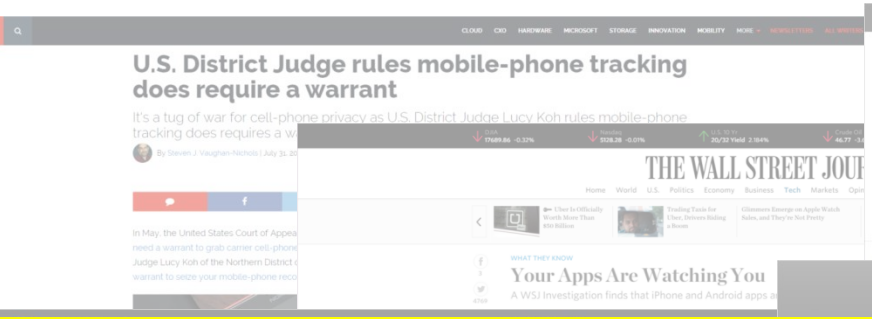
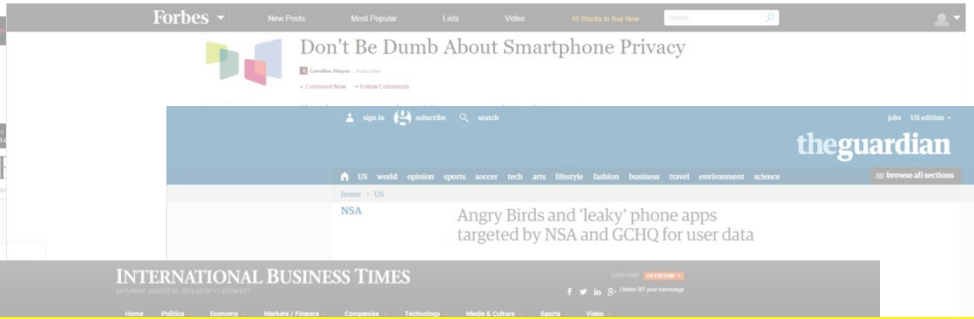
Anatomization and Protection of Mobile Apps' Location Privacy Threats

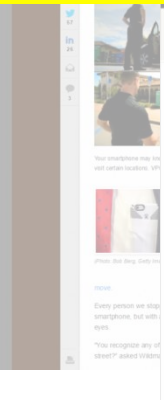
Kassem Fawaz, Huan Feng, and Kang G. Shin
Computer Science & Engineering
The University of Michigan

Apps' Location Privacy

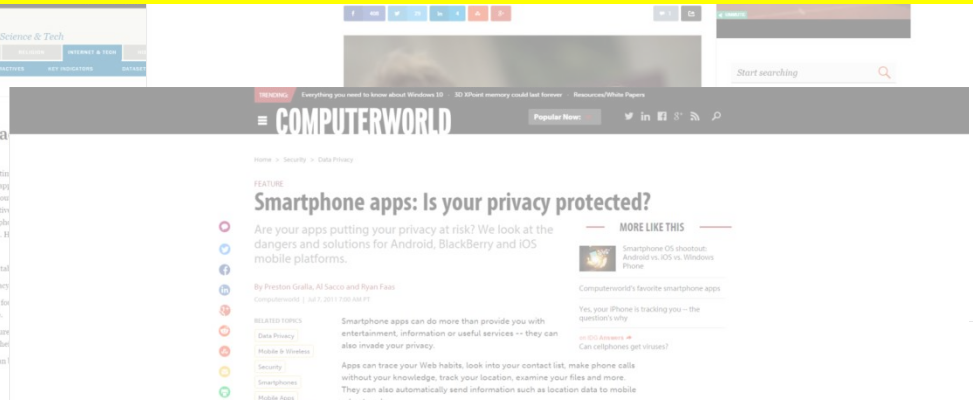


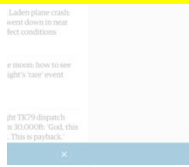


What can we do about it?





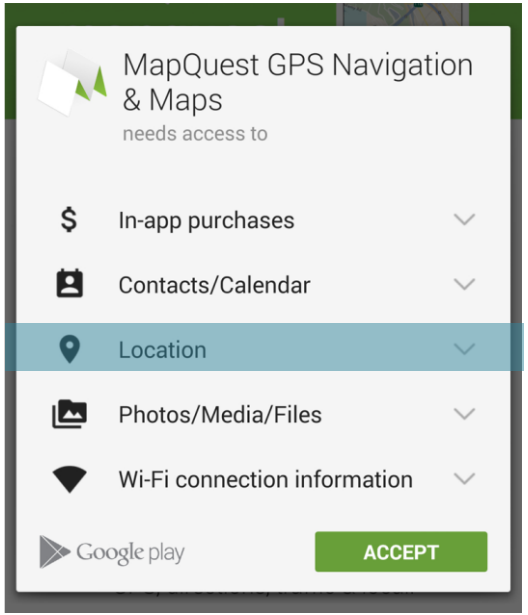


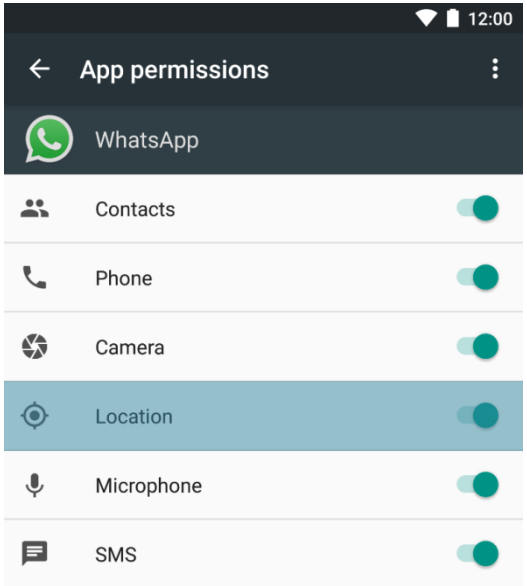


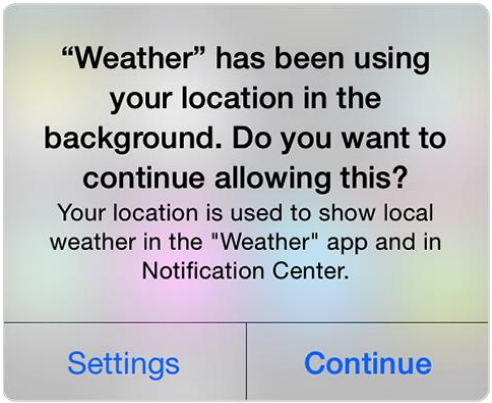
Apps' Location Privacy

- 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

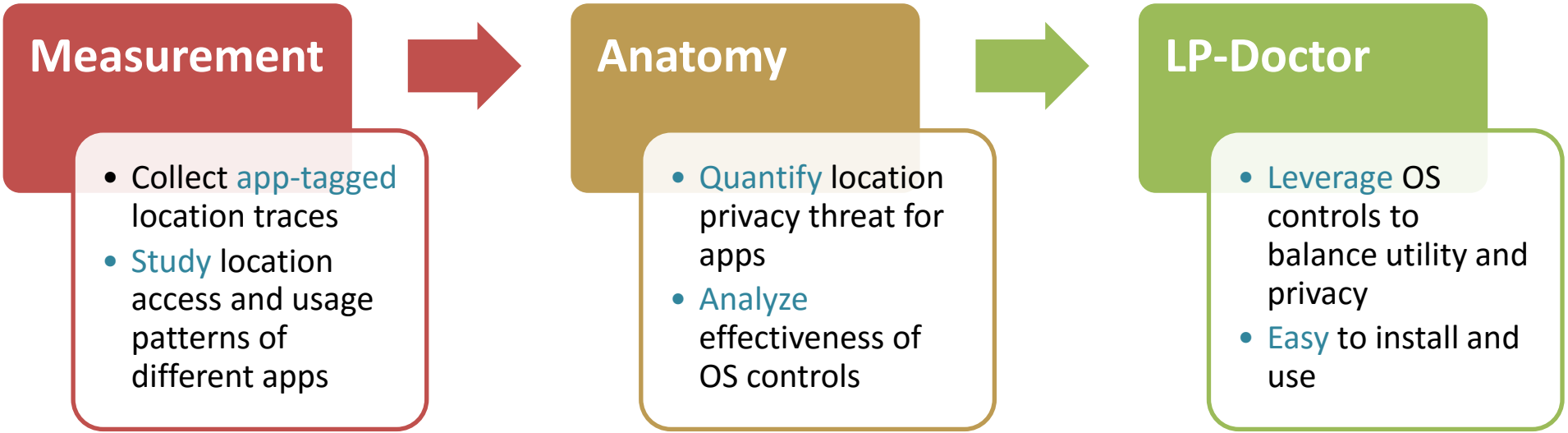
Background location access

http://arstechnica.com/gadgets/2015/05/android-m-dev-preview-launches-permission-controls-fingerprint-api-and-more/
https://support.apple.com/library/content/dam/edam/applecare/images/en_US/applecare/location_background.png

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



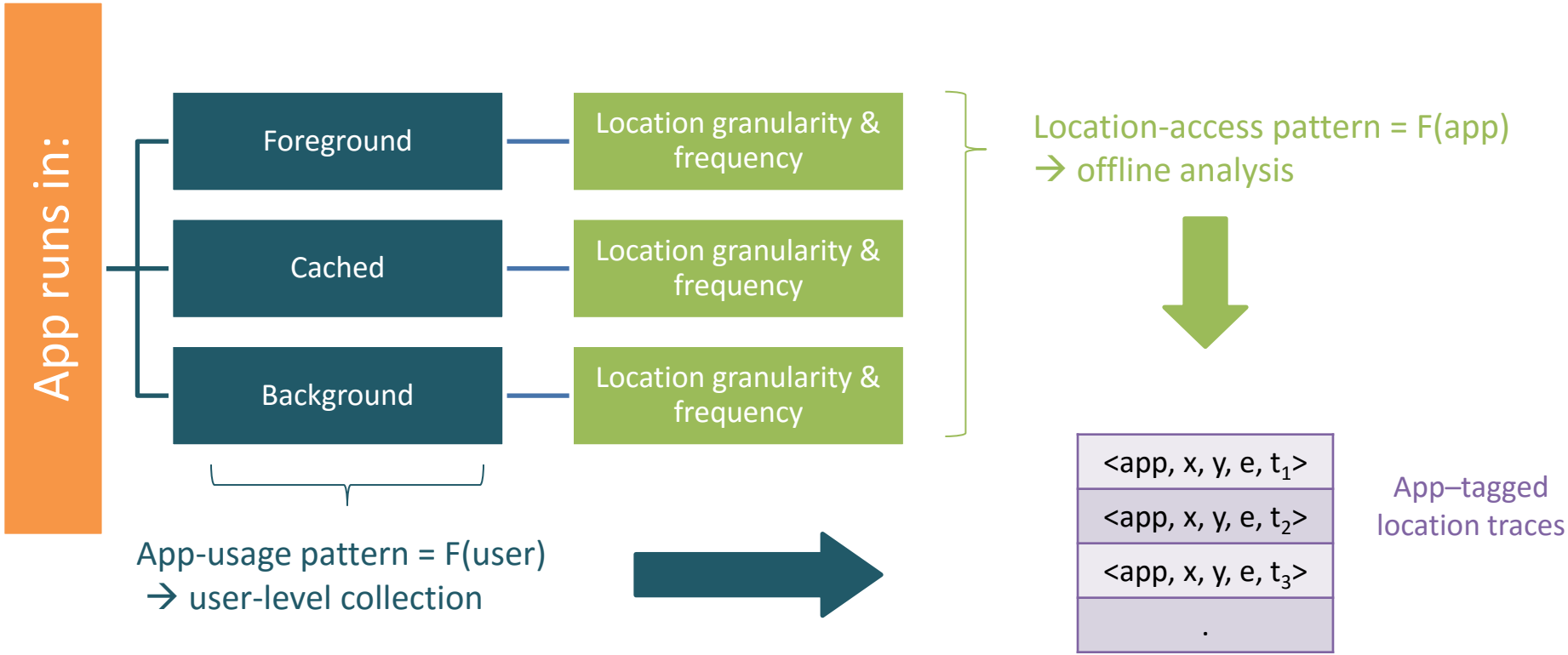
Measurement and LP-Doctor are based on Android, but conclusions apply to other platforms

Measurement



https://thisisprable.files.wordpress.com/2014/05/tape-measure1.jpg

Measurement



Location-Access Patterns

Location permissions	# apps	Foreground (%)	Cached (%)	Background (%)	None (%)	Coarse Granularity enough (%)
Coarse	210	71	6	1	22	100
Fine	955	74	14	4	12	48
All	1165	74	12	3	14	66

- 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
```

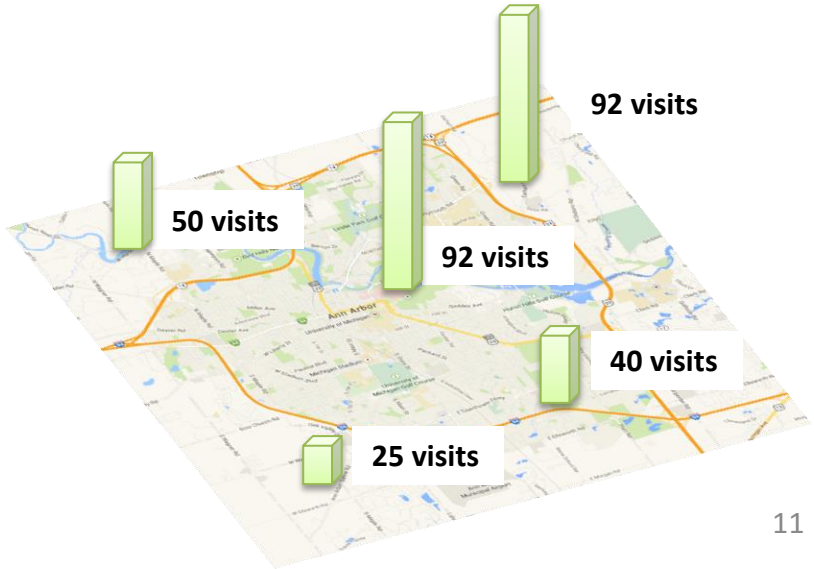
- Three datasets:

	RTCL	PhoneLab	LiveLab
Participants	25	95	30
Period	1 week - 10 months	1 week - 4 months	1 year
Location	Ann Arbor, MI	Buffalo, NY	Houston, TX

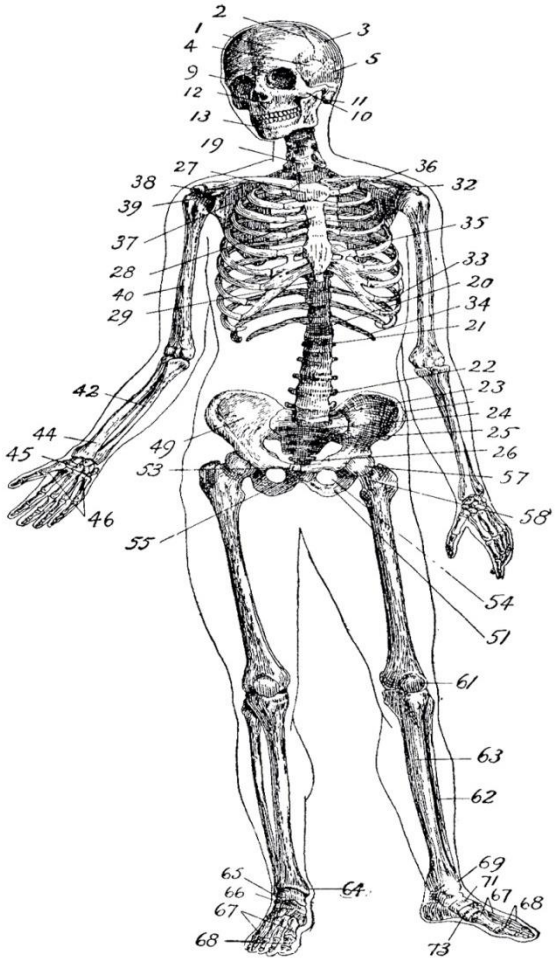
Location-Access Model

- 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



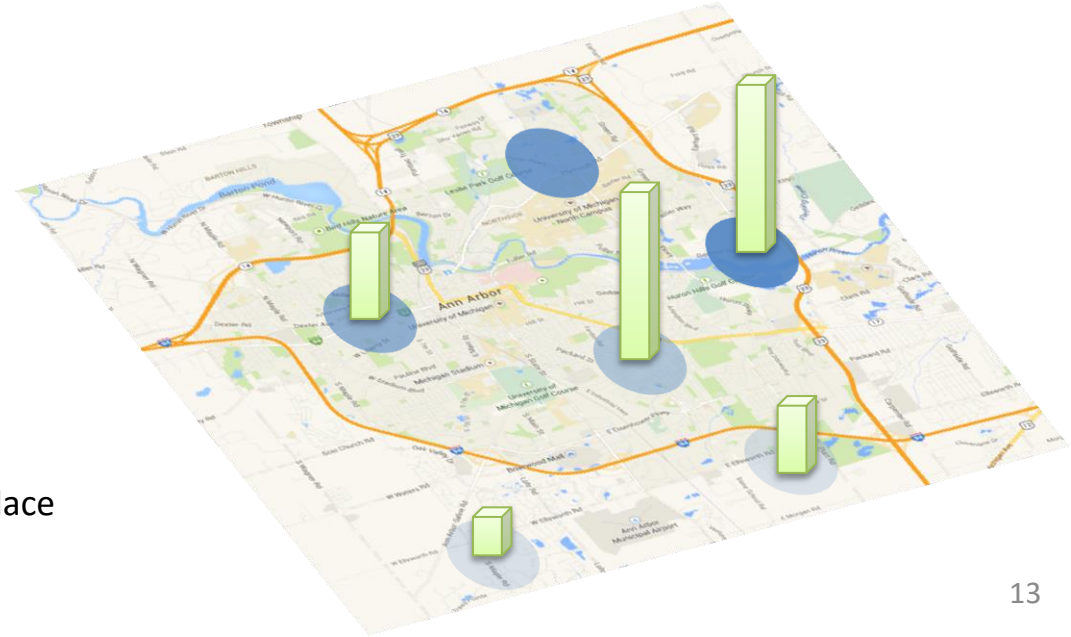
Anatomy



http://thegraphicsfairy.com/wp-content/uploads/2013/09/Vintage-Anatomy-Skeleton-Images-GraphicsFairy.jpg

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:
  - PoI_{total}: Portion of user's identified significant places
    - The closer to 1, the more places are identified
  - PoI_{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
  - $Prof_{bin}$: χ^2 test with significance level of 0.05
    - Whether the histogram (sample) is a good fit of the mobility pattern

Threat Anatomy

Metric	Insight

Threat Anatomy

Metric	Insight
PoI_{total}	• For each user: 20% of apps, each, identified most PoIs • Most apps identified users' top two PoIs

Threat Anatomy

Metric	Insight
PoI_{total}	• For each user: 20% of apps, each, identified most PoIs • Most apps identified users' top two PoIs
PoI_{sens}	• For each user: 20% of apps identified most sensitive PoIs • A&A libraries identified more of the sensitive PoIs

Threat Anatomy

Metric	Insight
PoI_{total}	• For each user: 20% of apps, each, identified most PoIs • Most apps identified users' top two PoIs
PoI_{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



http://site.ambercity.com/images/mabis/legacy-sprague-lc-rappaport-type-stethoscope-adult-black-10-420-020-lr.jpg

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

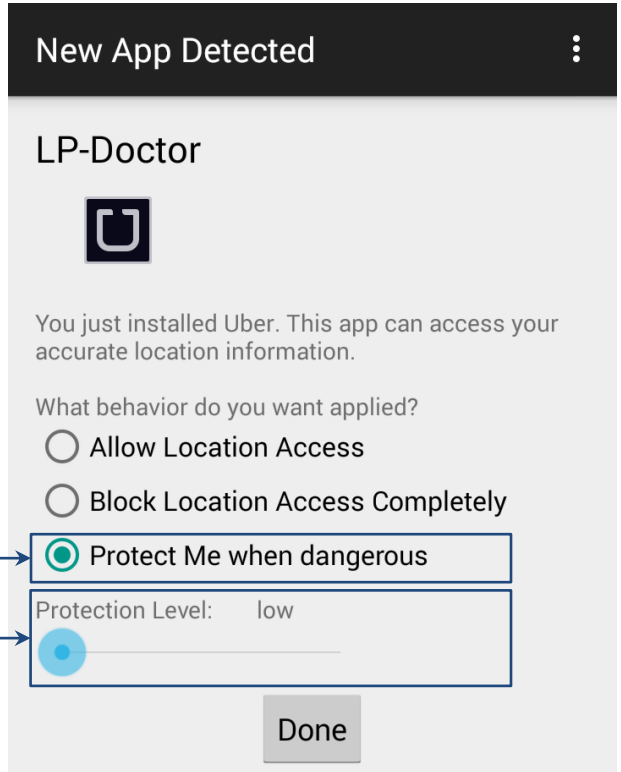
- What's in?
  - Honest-but-curious adversaries
    - Service providers or Advertisement and Analytics (A&A) agencies
    - Access location only through location APIs
    - Foreground location access
- What's out?
  - Background location access
    - Some platforms are providing controls like iOS
  - 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?

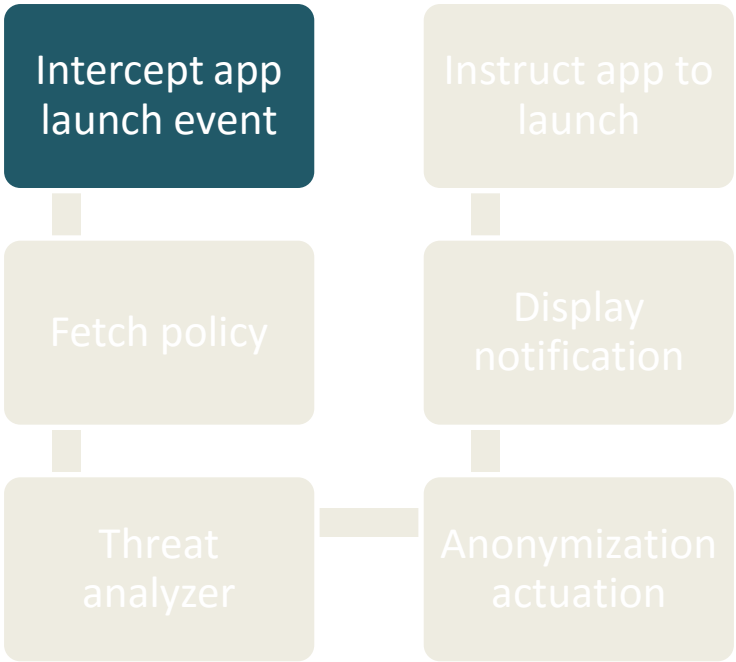
Apply protection before each session

Level decides:
- Amount of noise added
- Sensitivity level for $prof_{bin}$



LP-Doctor

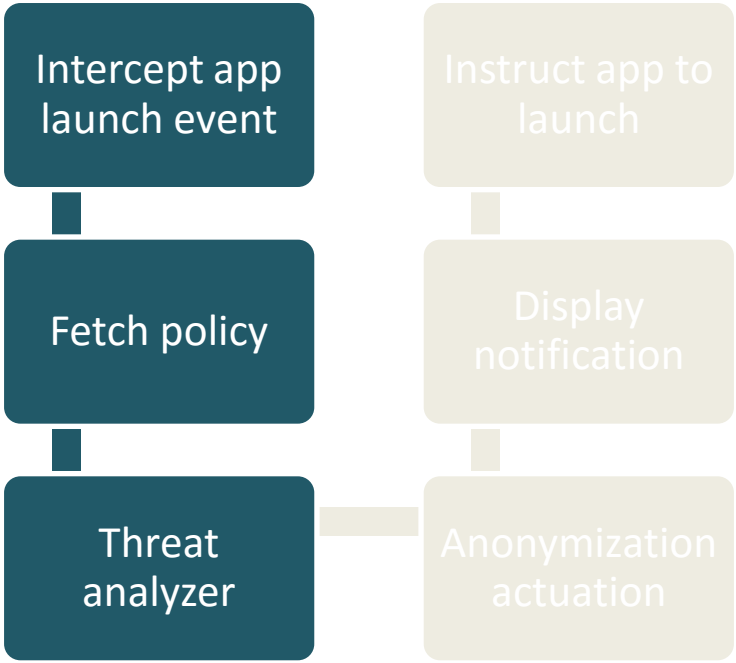
What happens when the user runs an app?

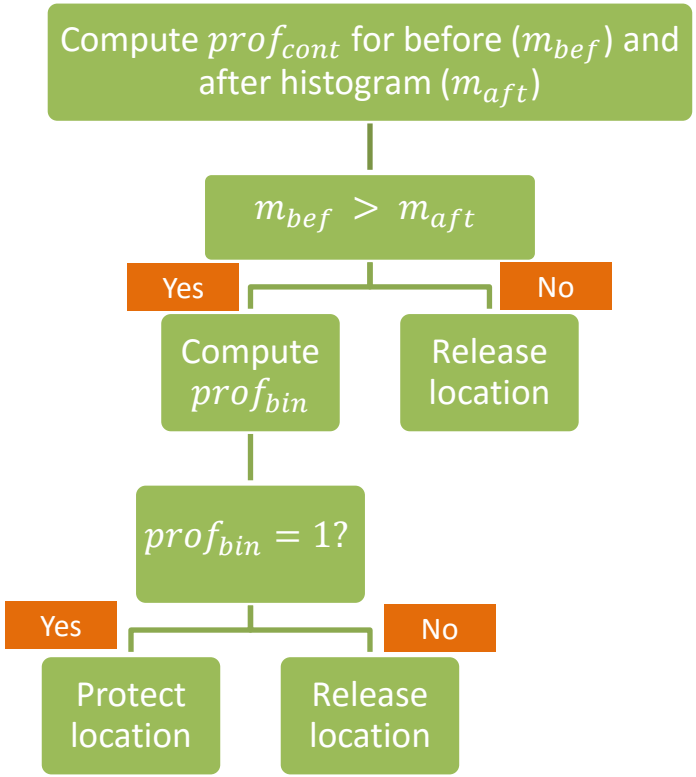


1. Instrument the CyanogenMod app launcher
2. Pause app execution
3. Divert execution to LP-Doctor

LP-Doctor

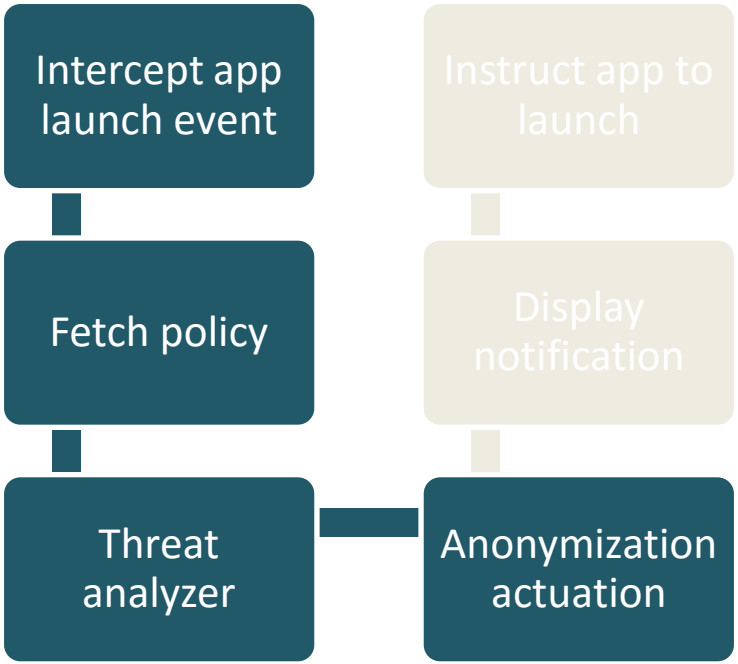
What happens when the user runs an app?





LP-Doctor

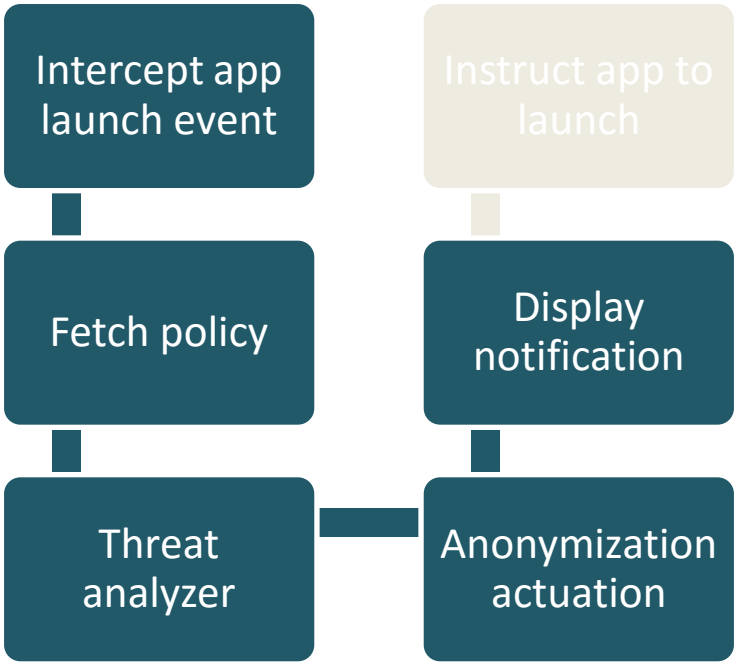
What happens when the user runs an app?

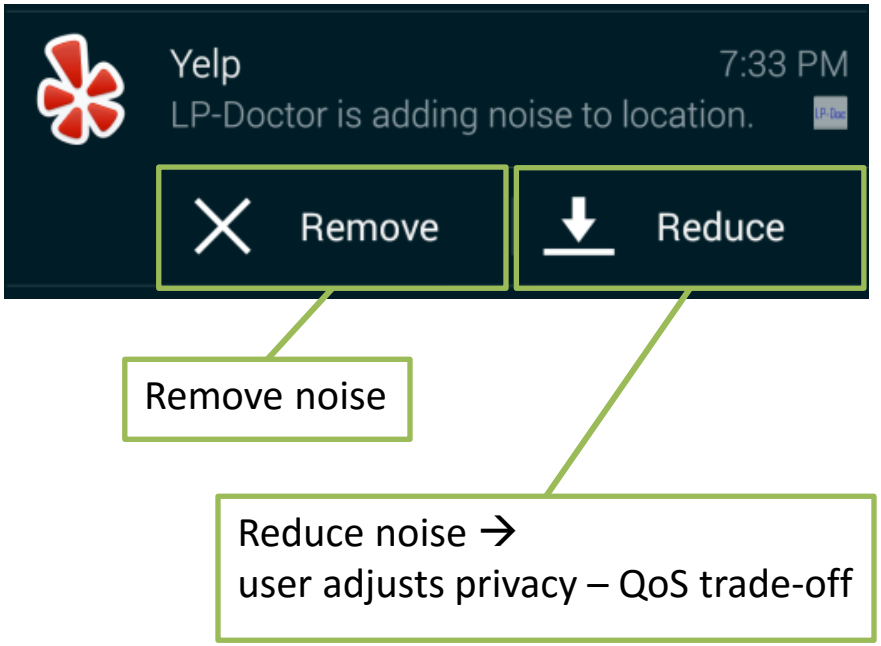


1. Compute obfuscated location
   - Add 2-D Laplacian noise that achieves indistinguishability
2. Engage Android's mock location provider
3. Apply same obfuscated location for future sessions of the same app from the place
   - Reduces information leaks

LP-Doctor

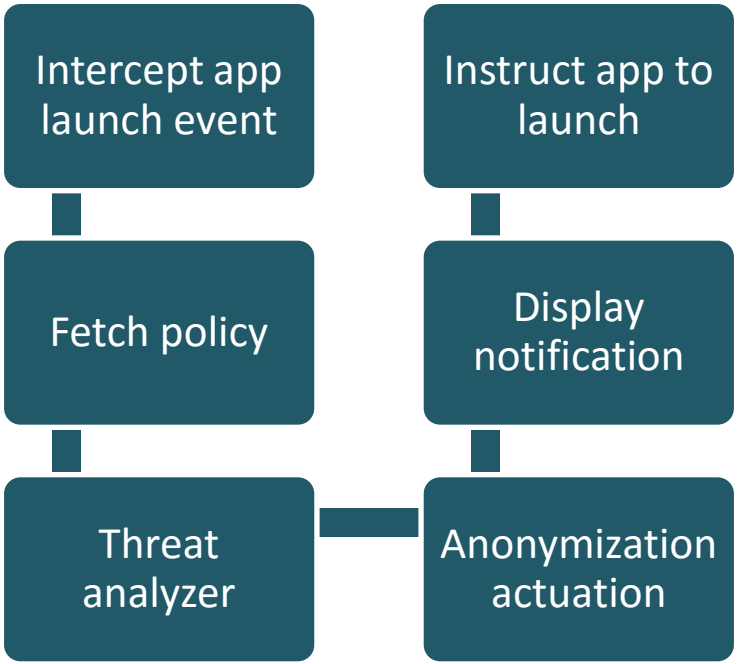
What happens when the user runs an app?





LP-Doctor

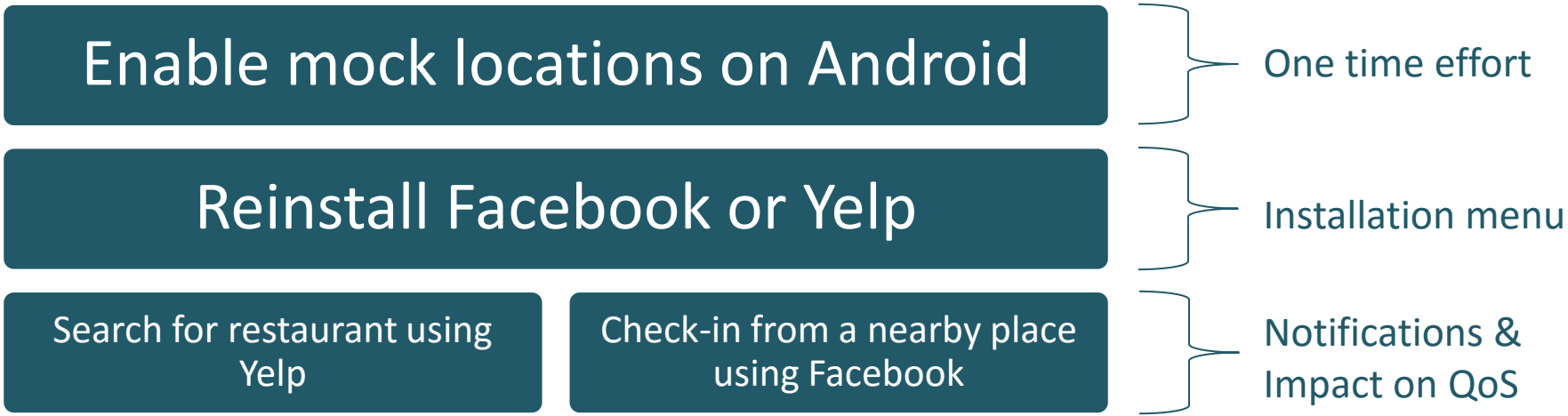
What happens when the user runs an app?



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

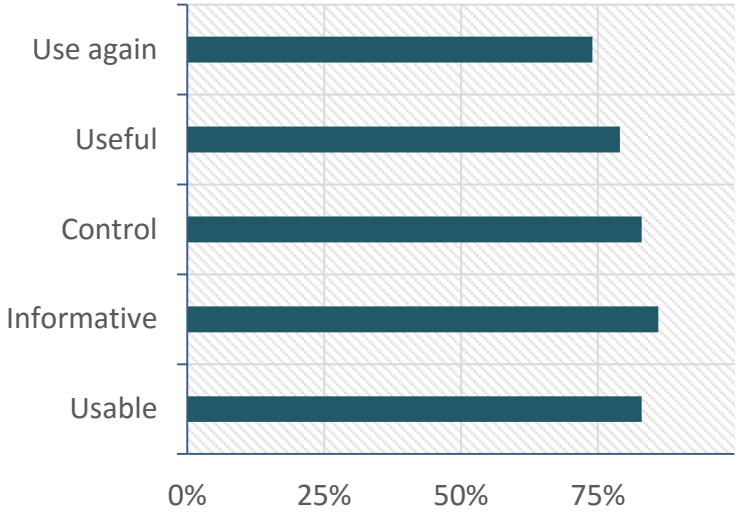
User Study

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



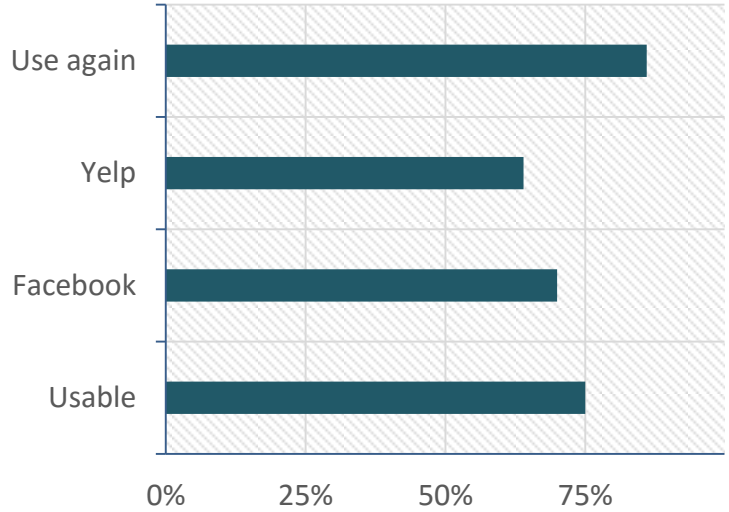
User Study

Installation Menu



Appears only once for each 5 installed apps

Notification

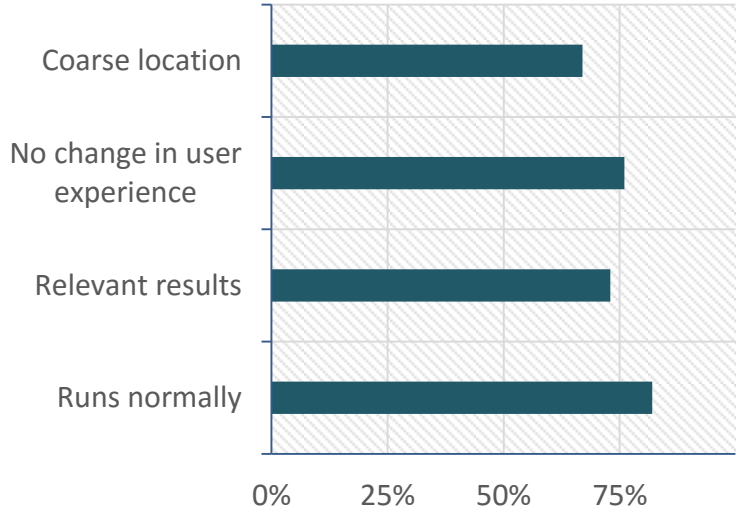


Appears for 12% of sessions on average

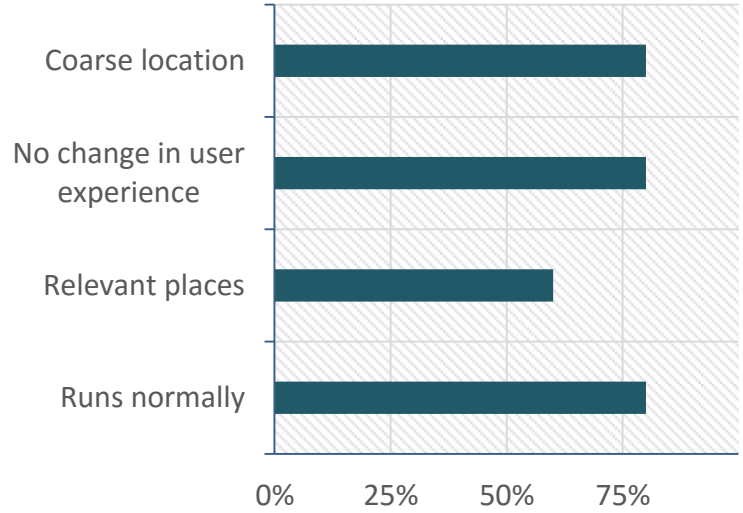
User Study

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

Yelp

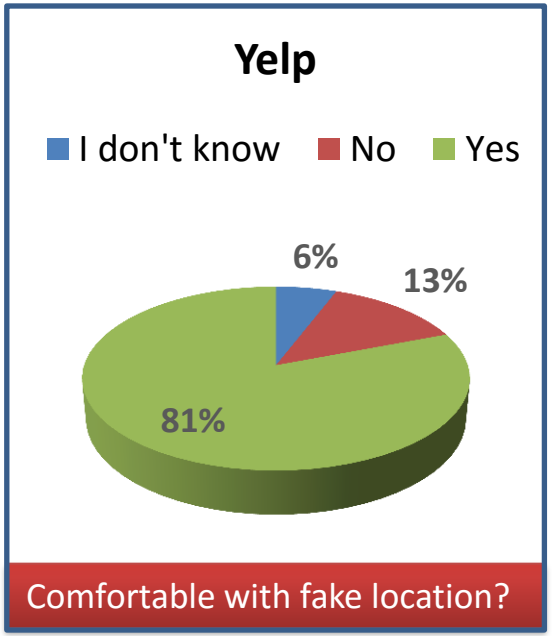


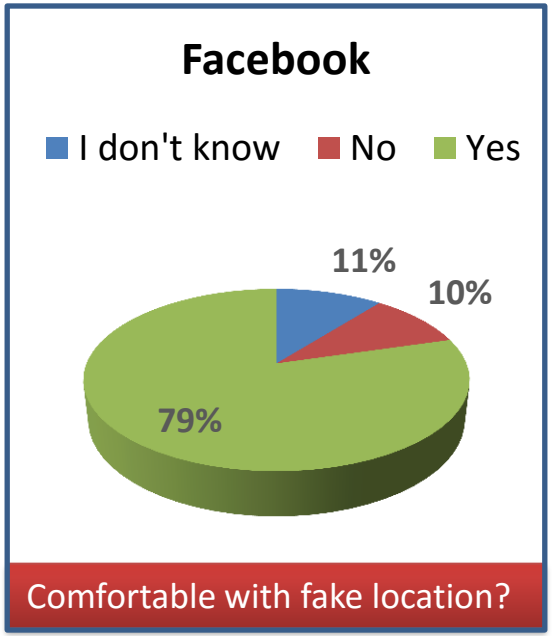
Facebook

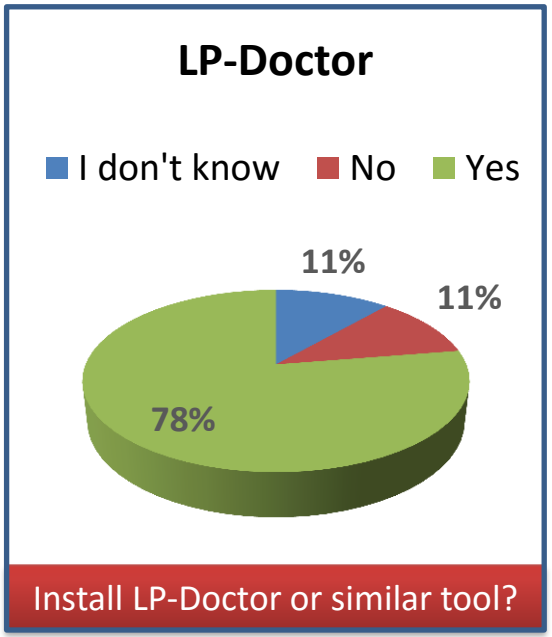


User Study

Post study questions







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