# Session 2: Mobile Privacy

PRIVACYCON

# ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic

David Choffnes, Northeastern University

**Co-authors**: Jingjing Ren (Northeastern University), Ashwin Rao (University of Helsinki), Martina Lindorfer (SBA Research), Arnaud Legout (Inria, Sophia Antipolis, France

This research was funded in part by the Data Transparency Lab

PRIVACYCON

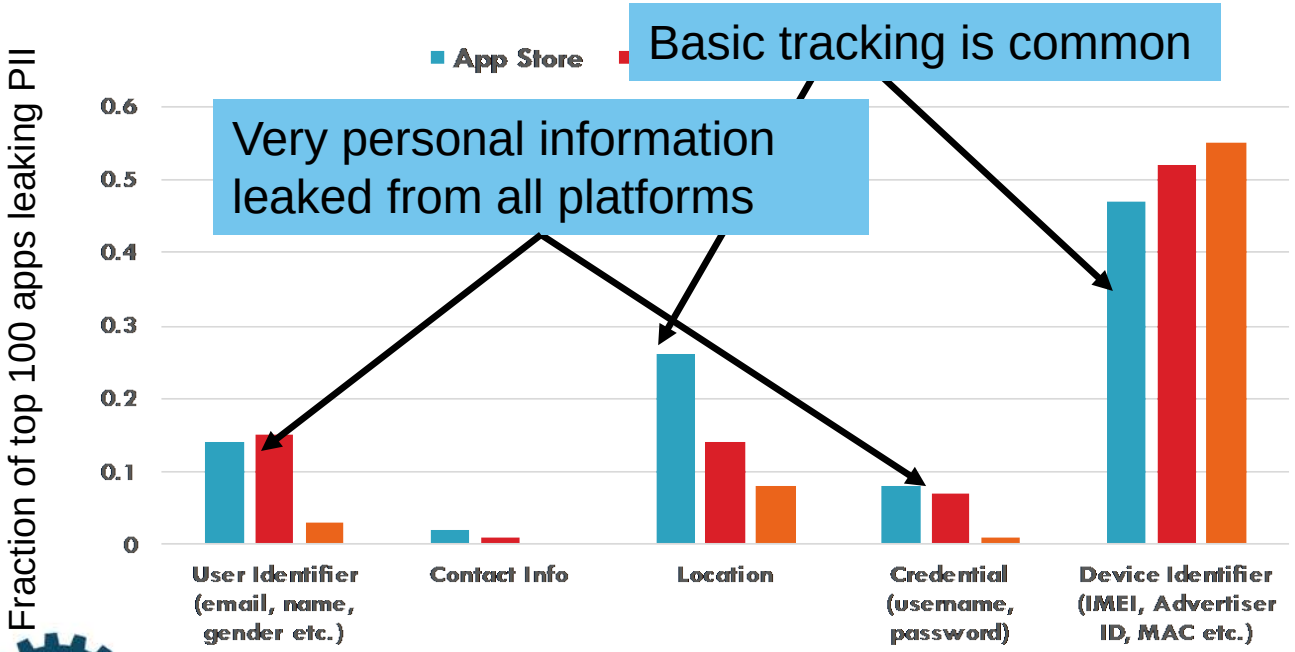
# Privacy in the mobile Internet

- Mobile vs. privacy
  - Rich sensors, ubiquitous Internet
  - Information sharing pervasive
- Users at a huge disadvantage
  - Poor control
  - Little visibility

# How big of a problem is this?

- Controlled experiments
  - Seed devices with conspicuous PII
  - **Manual tests** of top 100 apps for each OS
    - **iOS**, Android, **Windows Phone**
    - (Note results have **significantly better coverage** than automated tests.)

# PII leakage is pervasive






(Tested in September, 2015)

# How do we improve mobile privacy?

- Better software *on the device*
  - OS: Difficult to deploy, doesn't handle user input
  - App store: Requires buy-in, not portable

- Look for PII leaks ***in the network***
  - Analysis can run anywhere
  - Trivially easy if you know what PII to search for…

# Automatically identifying PII leaks

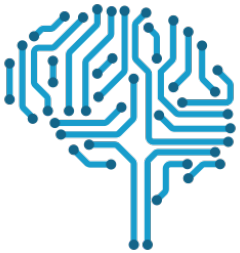
- Hypothesis: PII leaks use predictable formats
  - E.g., "`name=Choffnes`" or "`zip=02115`"

- Approach: ***Learn*** the format of PII leaks
  - Does not require knowing PII in advance
  - Resilient to changes in PII leak formats over time

# ReCon

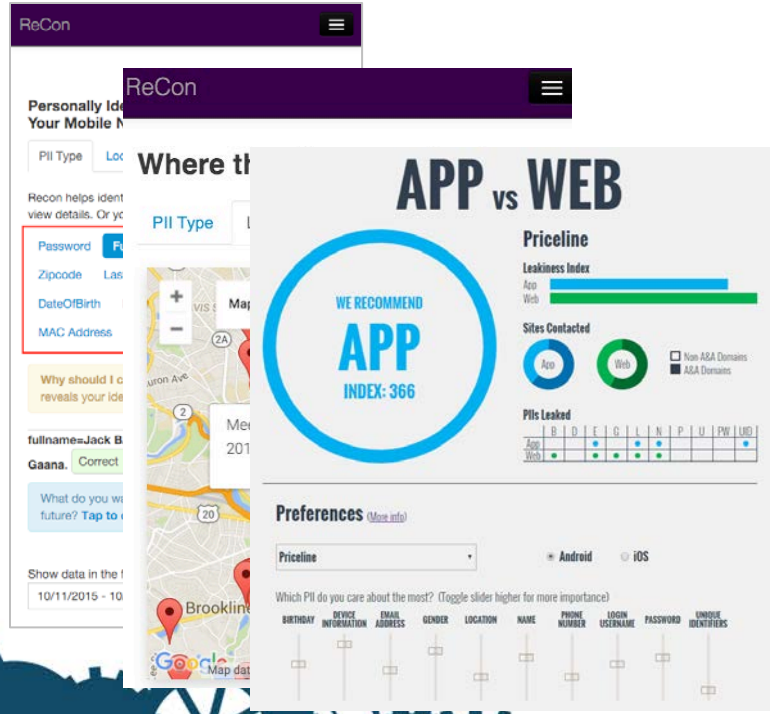
Network Trace Analysis

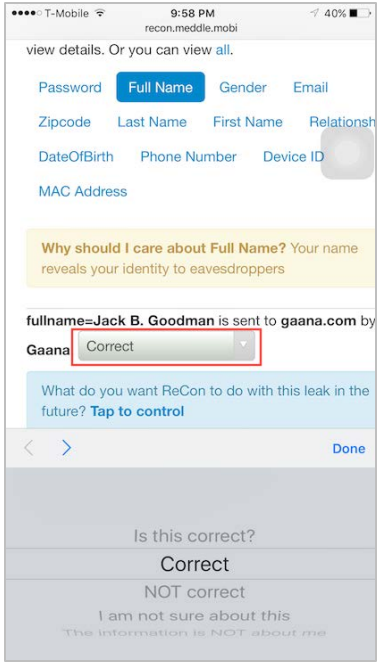

Machine Learning

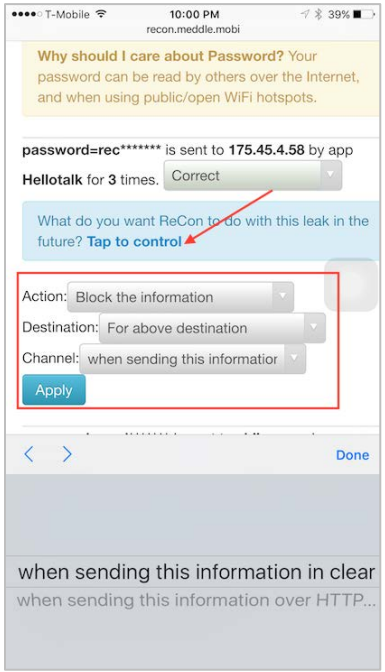


CrowdSourcing

# Revealing and controlling leaks



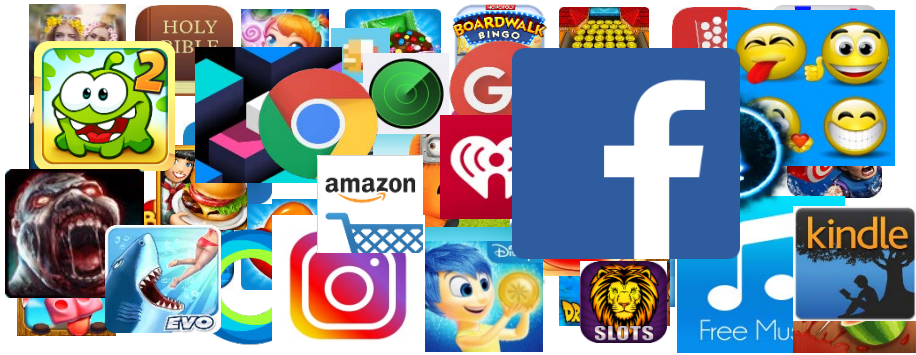
# Key results: Accuracy

- How accurate is ReCon?
  - **99% overall accuracy** from controlled experiments
  - FPR: 2.2%, FNR: 3.5%

# Key Results: User study

- IRB-approved user study (382 users as of November, 2016)
  - 220 iOS, 197 Android devices
  - 20/26 responses: system useful & behavior change
  - PII found: 27,009 cases (12,318 confirmed)

- Some details
  - ***199 cases of credential leaks***
  - Average leaks: iOS > Android
  - Unexpected, suspicious leaks
    - Recipe/cooking app tracks location
    - Video/Game/News app leaks gender



# Impact: security and transparency

- *Identified 25 apps exposing **passwords,** (most) in **plaintext***
  - Used by **millions** (Match, Epocrates)
  - Responsibly disclosed
  - Many have not fixed the problem

- *Interesting responses from developers*
  - “Thank you for responsibly disclosing this”
  - “We do not claim to be a secure messaging app”
  - “Sending passwords in plaintext is intentional”

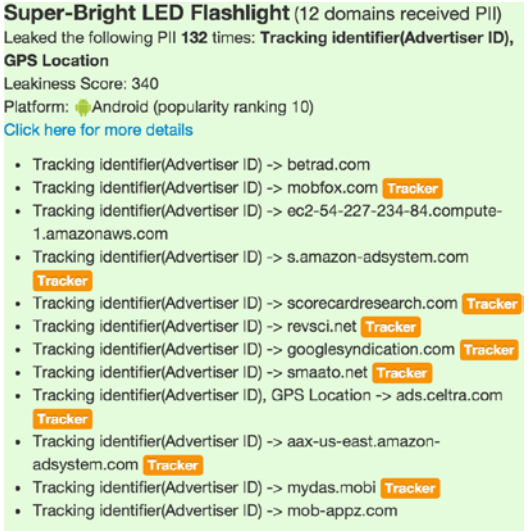
# Naming names and listing leaks

- Anonymized report of app PII leaks

https://recon.meddle.mobi/app-report.html

- Same for mobile Web browsing

https://recon.meddle.mobi/web-report.html



# Wrapup

- ReCon improves **transparency** and **control** over PII
  - **Learn** what information is being leaked
  - **Crowdsourcing** to determine correctness/importance
  - Allow users to **block/change** what is leaked
- Ongoing/future work
  - Deploying on devices (Haystack/Lumen), routers
  - Applying analysis to IoT devices

# Acknowledgements


- Contributors

  **Jingjing Ren (NEU)**, Martina Lindorfer (UCSB), Ashwin Rao (U. Helsinki), Arnaud Legout (INRIA)


**Sign up at** https://recon.meddle.mobi

# Understanding the Mobile Ecosystem with the Lumen Privacy Monitor

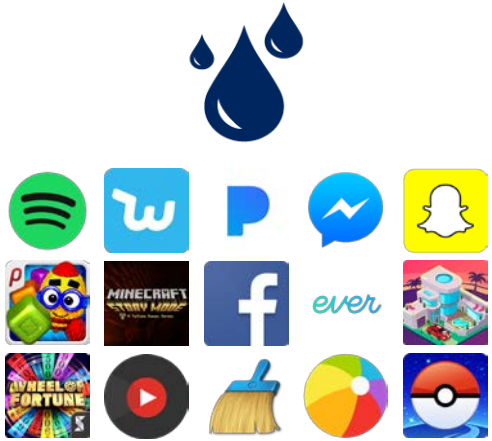
**Narseo Vallina-Rodriguez**

Project sponsors:

NSF

DATA TRANSPARENCY LAB

**DISCLAIMER:** The opinions expressed in this presentation are solely those of the presenter



1st parties (Direct)



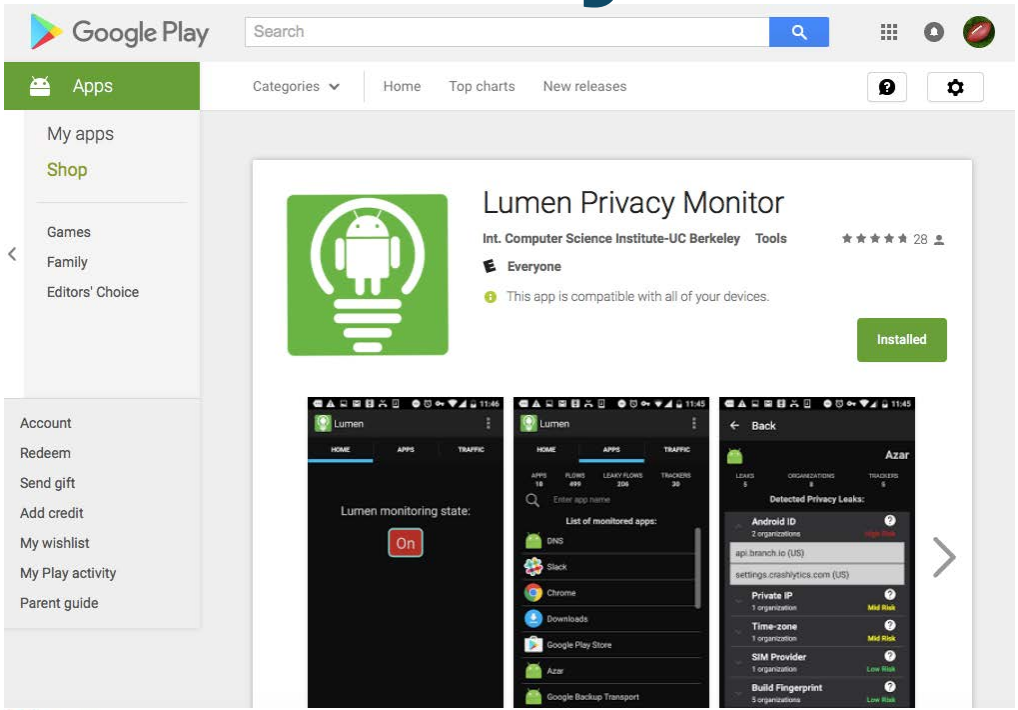
3rd parties (Indirect)

# Project Goals

- Identify 3rd-party tracking services on mobile apps
- Evaluate their impact on user privacy
- Promote mobile transparency and enable user control

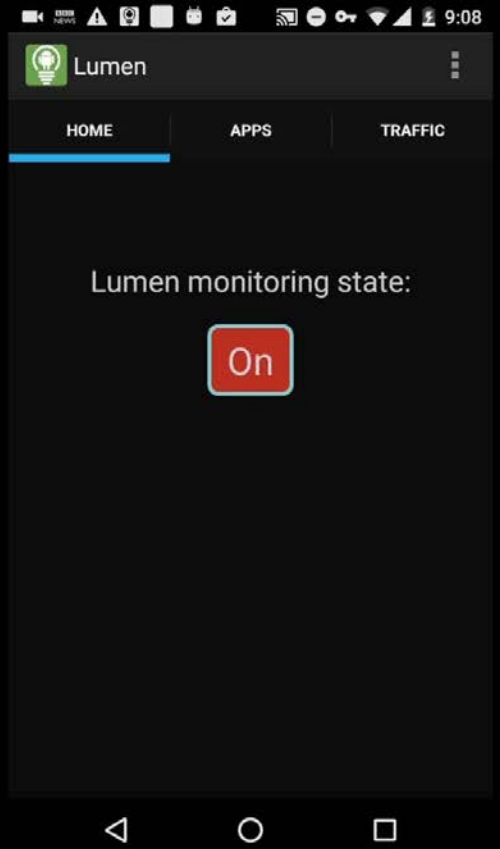
# How?

# Lumen Privacy Monitor




PRIVACYCON

# Lumen Privacy Monitor



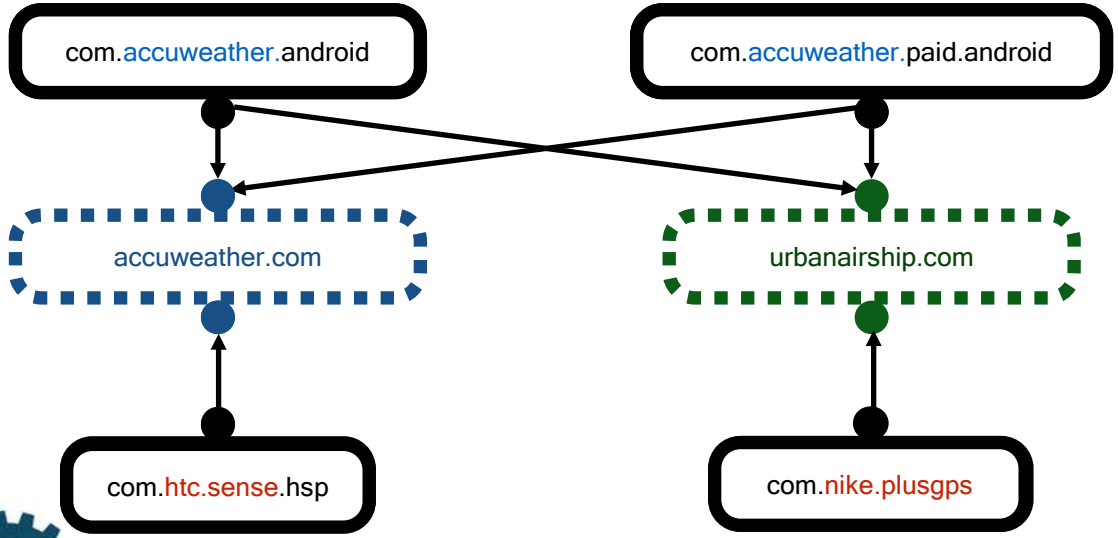
# Preliminary research results

# Dataset

- Accurate traffic fingerprints
  - 1000+ users (containing real user-stimuli)
  - 2,900+ apps
  - 3,200+ second-level domains

# 1st party vs. 3rd party services

Basic heuristic: *deg* (n)>1



How to distinguish ad networks and trackers (ATS) from CDNs and other online services?

PRIVACYCON

# Challenges in classifying domains

1. Domain blacklists (e.g., Easylist) are web-oriented
2. URL classification services are inaccurate and incomplete

| | URL | Status | Categorization | Reputation |
|---|---|---|---|---|
| | http://flurry.com | Categorized URL | - Internet Services | Minimal Risk |

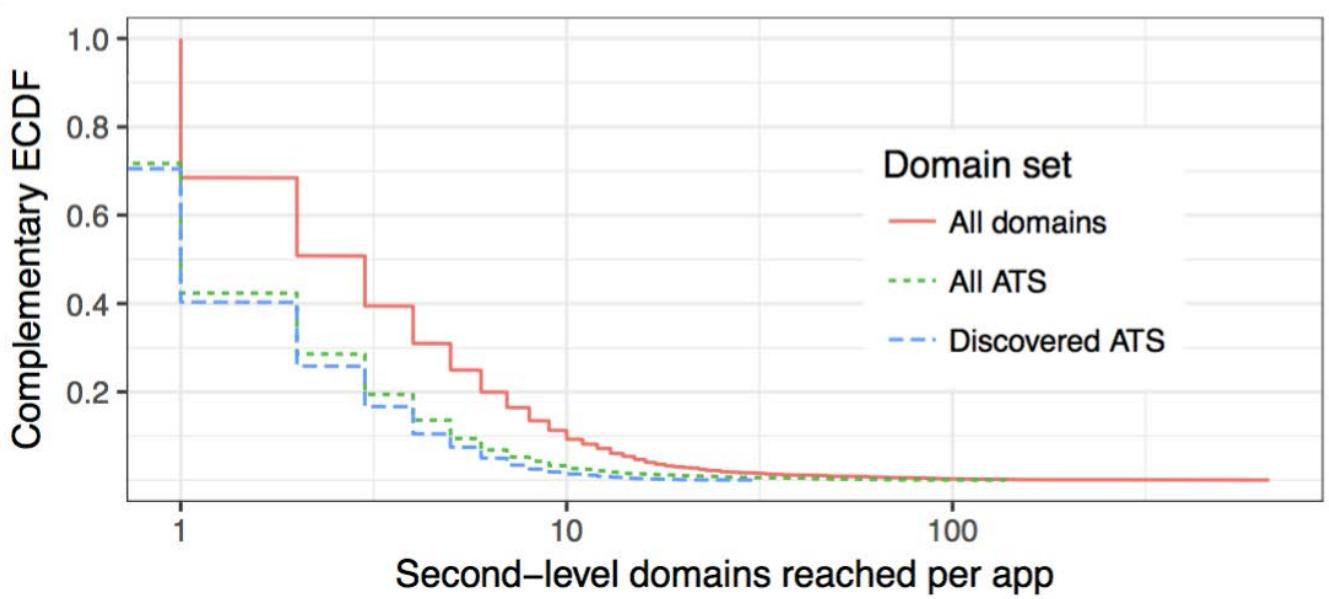
# Custom classifier

- Identify unique identifiers in data flows
  - IMEI, IMSI, Android ID, MAC Address, Serial Number, …
- Analyse the content of their landing pages with a web scrapper and NLP

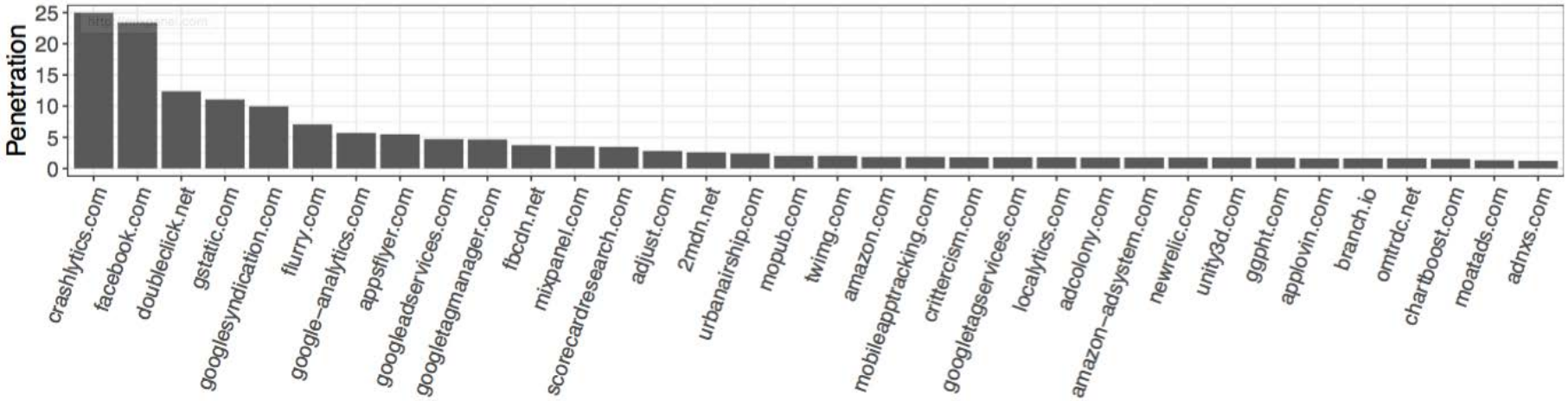
# Results

| Set | Total | Previously reported ATS (%) | Third Parties (%) |
|---|---|---|---|
| All domains | 3261 | 11 | 31 |
| UID Harvesters | 336 | 9 | **41** |

# Ad and Tracking Services (ATS) in mobile apps

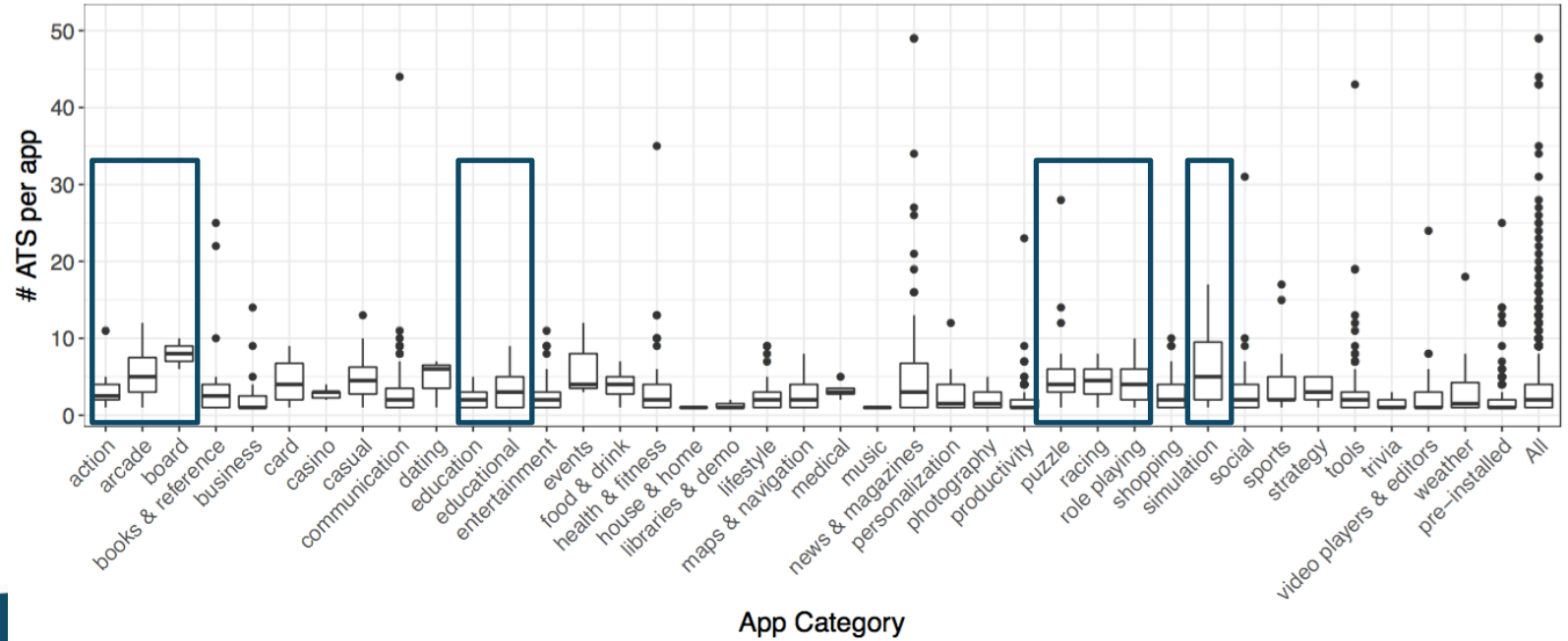


# Service popularity




Over 68% of tracking services are cross-platform

# Trackers by app category




Are games and educational apps COPPA compliant?

# Abusive practices (I)

- `getprop` command contains unique IDs
- Unprotected by Android permissions
  - Enables tracking without user consent

```
[dhcp.wlan0.domain]: [networks.imdea.org]
[net.hostname]: [android-db216281e95dfab1]
[persist.service.bdroid.bdaddr]: [40:B0:FA:5C:D0:80]
[ro.boot.serialno]: [04efb34e55e22fcc]
[ro.build.fingerprint]: [google/occam/mako:5.1.1/LMY48T/2237560:user/release-keys]
```

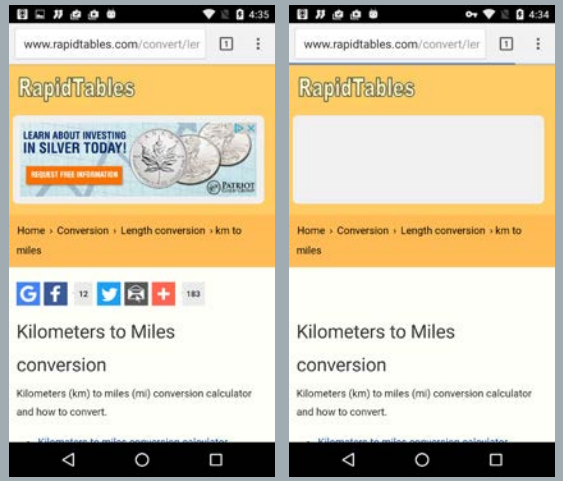
# Abusive practices (II)

Host: **track.XXXX.com**
Accept-Encoding: gzip

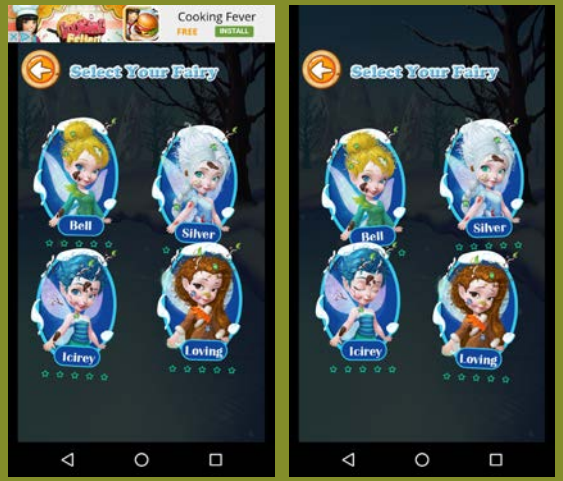
device=angler&installDate=2016-11-02_0126-0700&firstLaunchDate=2016-11-02_0126-0700&sdk=23&carrier=&date1=2016-11-02_0126-0700&af_preinstalled=false&***advertiserIdEnabled=false***&TRACKERKey=yZnL9BNtUzZLva6evLpUg5&lang=English&app_version_name=2.2.0&dkh=yZnL9BNt&**android_id=84f942c74fffbdef**&**advertiserId=fff3ca7e-61d7-4298ab14-256033002de9**&deviceType=userdebug&af_v=da33e2cb0879238eb1dc9d93e0ce38b4564fbd9d&app_version_code=3&network=WIFI&operator=&brand=Android&date2=2016-11-02_0126-0700&af_timestamp=1478118372355&uid=1478118365655-13890785443306038 68&isFirstCall=true&counter=1&product=aosp_angler&model=AOSP+on+angler

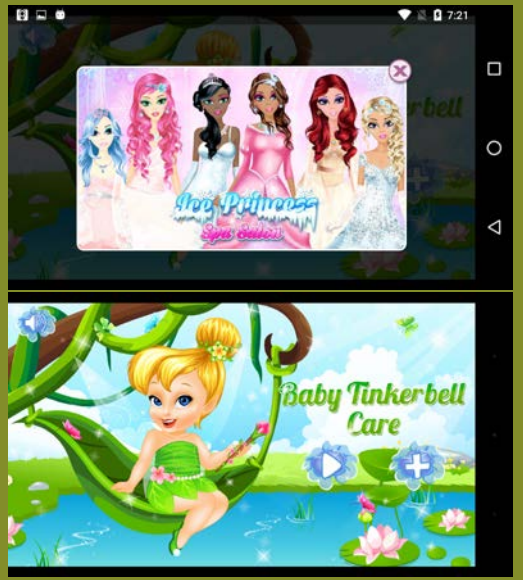
# A tool for users

# Enabling system-wide user control




Web

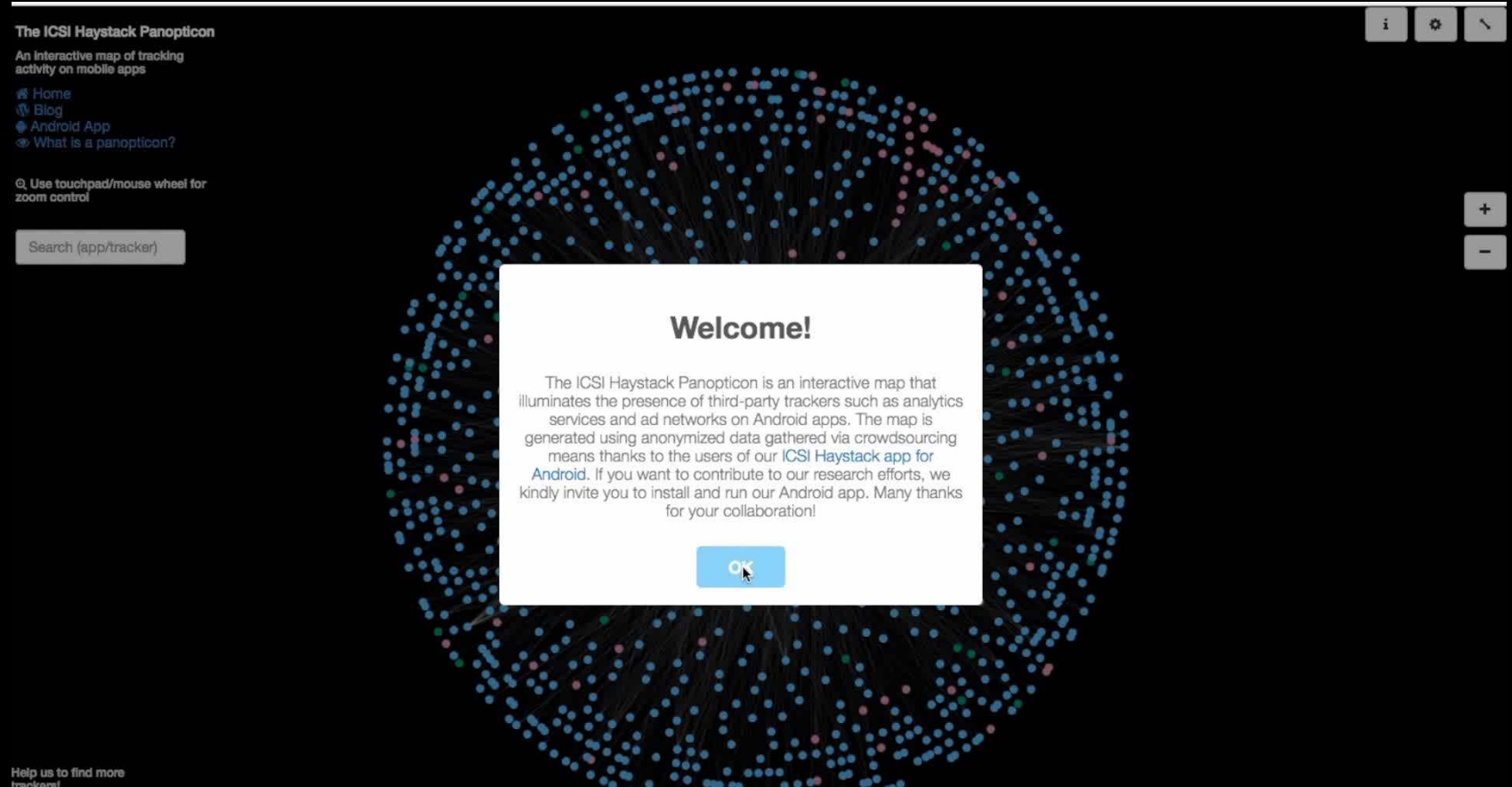



Apps

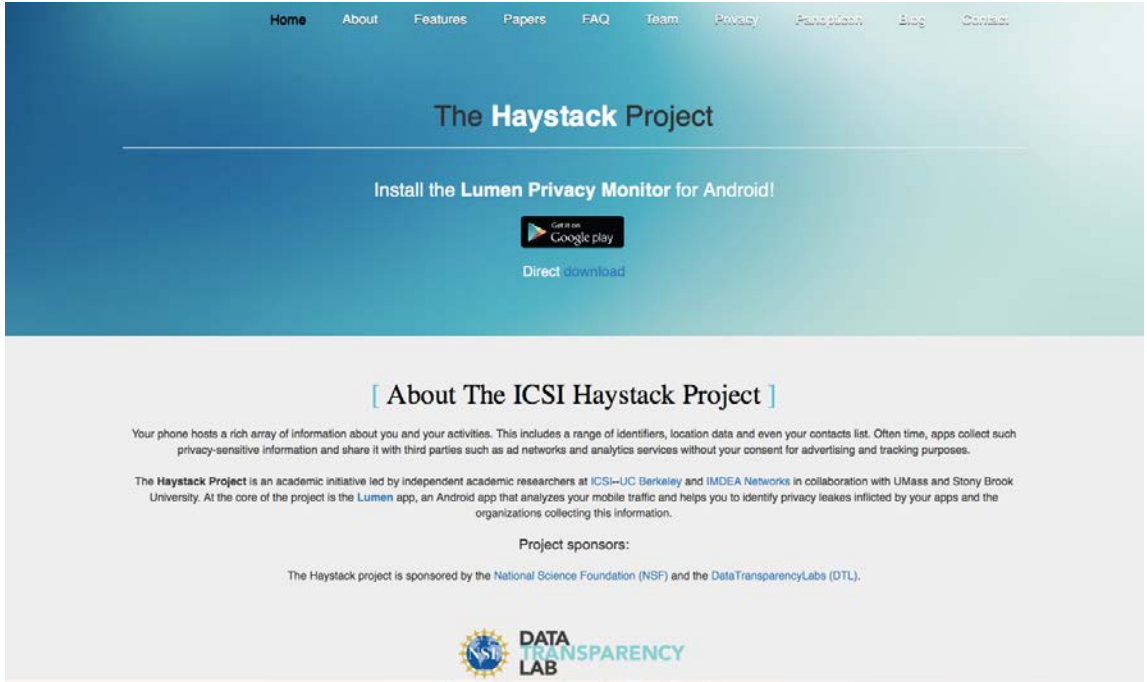
PRIVACYCON

# https://www.haystack.mobi/panopticon



# Q&A

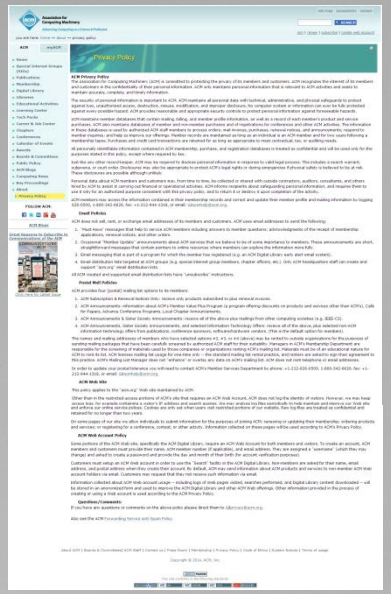
narseo@icsi.berkeley.edu
https://www.haystack.mobi



# Automated Analysis of Privacy Requirements for Mobile Apps

Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Peter Story, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven M. Bellovin, Joel Reidenberg

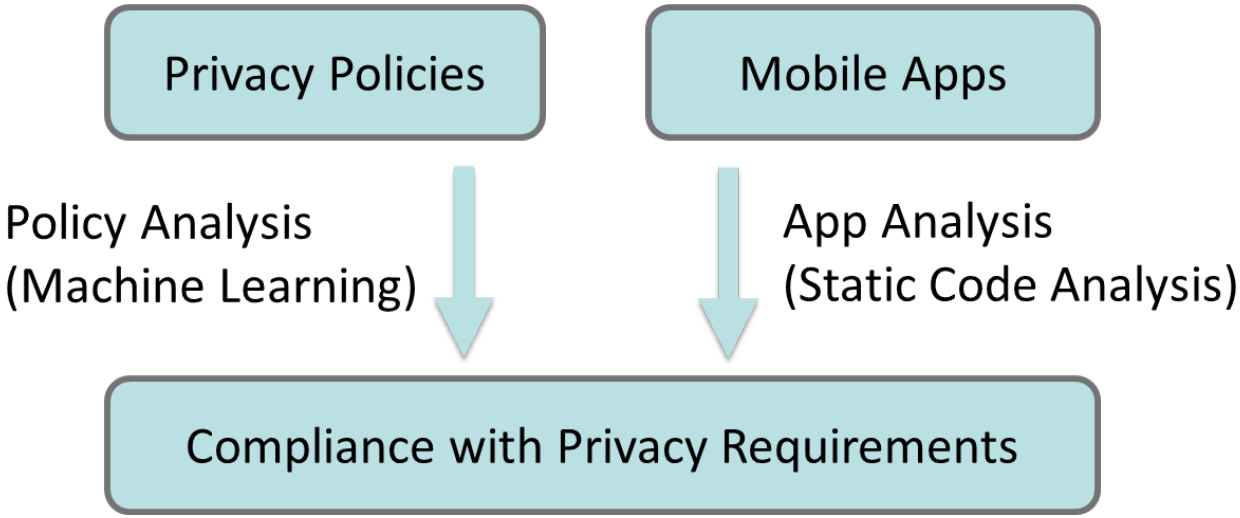
The research presented was funded by the National Science Foundation, the Defense Advanced Research Projects Agency, and the Air Force Research Laboratory



- Device IDs
- Location
- E-Mail
- ...

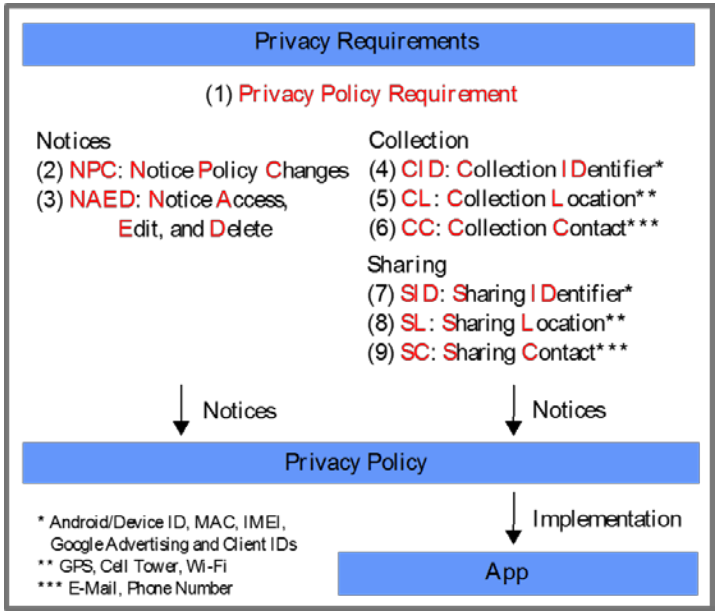
# Privacy Requirement Compliance?

# Policy and App Analysis



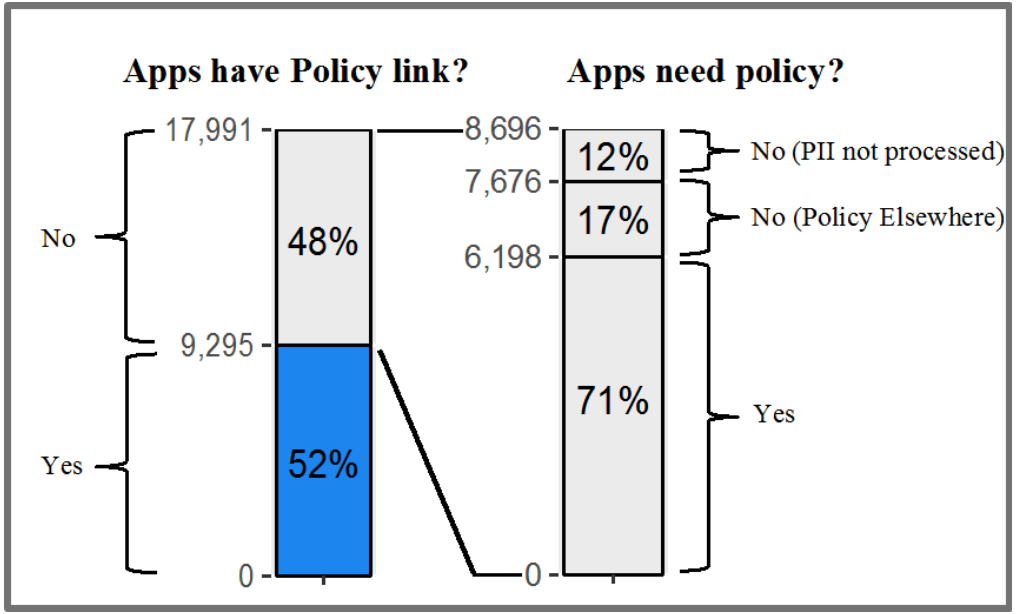
# Privacy Requirements

- Privacy requirements are <u>self-defined standards</u> (derived from privacy laws)
- Privacy requirement inconsist-encies are not necessarily violations of the law



# Many Apps do not have a Policy

- 71% of apps that appear to lack a privacy policy should have one



# Policy Analysis Method

Classify practices based on machine learning

1. Sentences in policies are extracted based on data type keywords (e.g., all sentences containing “location”)
2. Using action keywords unigram and bigram feature vectors are constructed from the extracted sentences (e.g., “share location”)
3. Apply support vector machines and logistic regression

# App Analysis Method

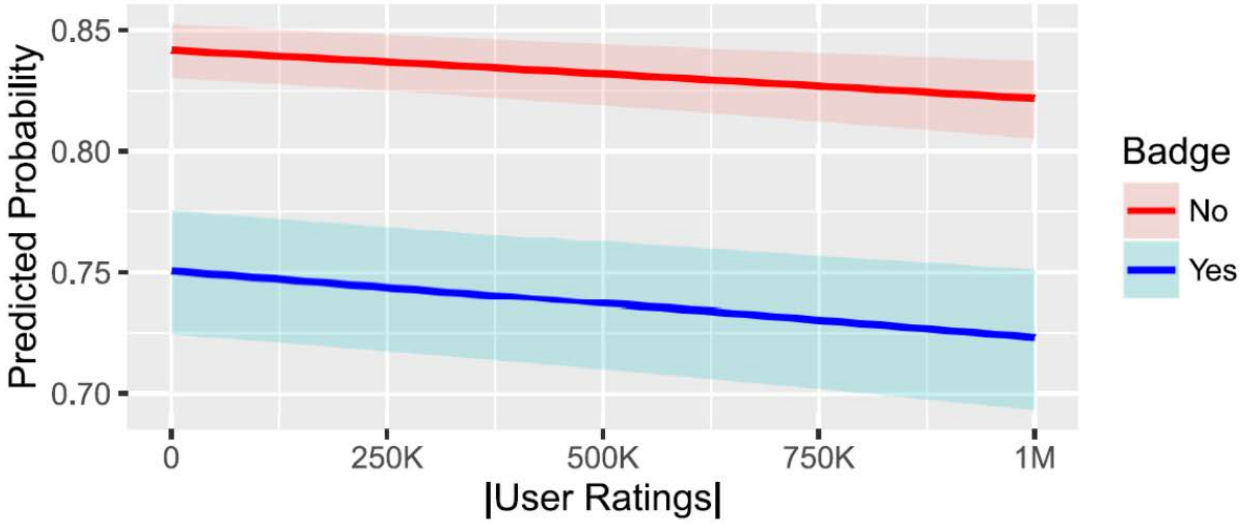
Analyze apps based on static code analysis

1. Identify relevant Android system and third party APIs
2. Perform static analysis on the app code (consisting of permission extraction, call graph creation, and call ID analysis)

# Results

| *Practice* | *$Prec_{pos}$ (n=40)* | *$Rec_{pos}$ (n=40)* | *$F\text{-}1_{pos}$ (n=40)* | *Inconsistency (n=9,050)* |
|---|---|---|---|---|
| CID | 0.75 | 1 | 0.86 | **50%** |
| CL | 0.54 | 1 | 0.7 | **41%** |
| CC | - | - | - | **9%** |
| SID | 0.93 | 0.74 | 0.82 | **63%** |
| SL | 1 | 1 | 1 | **17%** |
| SC | 1 | 1 | 1 | **2%** |

# Results for Sets of Apps



# Future Work

- Collaboration w/ the California Office of the Attorney General
- Help regulators, app store owners, developers, activists, …
- Extension towards websites, iOS apps, Internet of Things …
- Integration into app stores, software development tools, …

# Primal Wijesekera

## *The Feasibility of Dynamically Granted Permissions: Aligning Mobile Privacy with User Preferences*

Co-authors: Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David Wagner (University of California, Berkeley); Konstantin Beznosov (University of British Columbia)

The research presented was funded by the U.S. Department of Homeland Security's Science and Technology Directorate, the National Science Foundation, and the Natural Sciences and Engineering Research Council of Canada

# the feasibility of dynamically granted permissions

Primal Wijesekera, UBC / UC Berkeley

PRIVACYCON

# ask-on-every-use?

213 requests per hour!

- location (10,960/day)
- reading SMS data (611/day)
- sending SMS (8/day)
- reading browser history (19/day)

asking each time is infeasible

…but 80% wanted to block at least one request

(on average, they wanted to block 35% of all requests)

# users should be prompted about privacy decisions…

*only* when they are likely to care

*only* when the system does not know user preferences

# users make contextual decisions

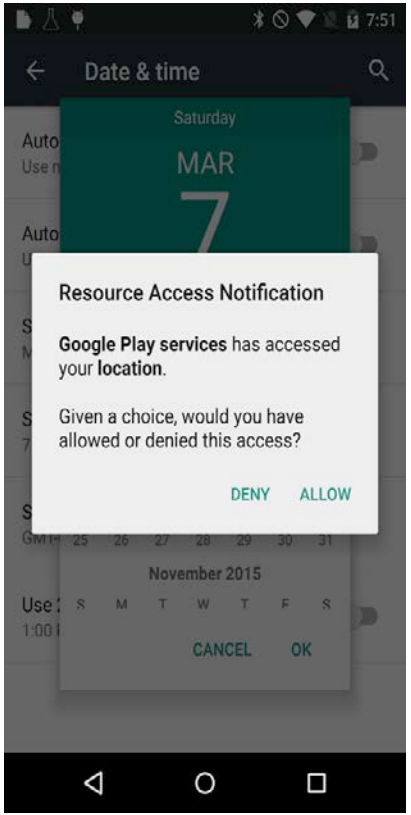
expectations drive privacy decisions

usage of an application is a strong contextual cue

# can we predict privacy decisions?

field study to collect behavioral data

probabilistic prompts to measure user expectations under different contexts



# the results

***133*** Android smartphone users

***176 million*** events recorded

***4,224*** prompt responses

# ask-on-first-use

an improvement over ask-on-install

didn’t match user expectations 15% of the time

# can we use machine learning?

## permission information

- permission
- visibility
- time of day

## user behavior

- browsing habits
- audio preferences
- screen locking habits

## contextual preferences

- under different visibility levels
- under different foreground applications

# contextual cues helped

| | Error Rate | Average Prompts/User |
|---|---|---|
| Ask-on-first-use | 15.4% | 12.34 |
| ML Model (behavior) | 24.9% | 00.00 |
| ML Model (contextual) | 03.2% | 12.00 |
| ML Model (contextual) | 07.4% | 08.00 |

# contextual privacy preferences...

were the most predictive feature group in predicting future privacy preferences

capture user's expectations under different contexts:

- foreground application
- visibility of the requesting application

# based on the confidence system can…

automatically allow access when a user is likely to expect it,

automatically deny access when a user is likely to not expect it,

prompt when system cannot infer user expectations (and learn from it)

# open questions

how can a system increase the transparency of automated decision-making?

how can passively observable traits be used to improve learning?

what's the best strategy to deny access?

# Discussion of Session 2

Presenters:

- **David Choffnes**, Northeastern University
- **Narseo Vallina-Rodriguez**, IMDEA Networks; International Computer Science Institute
- **Sebastian Zimmeck**, Carnegie Mellon University
- **Primal Wijesekera**, University of British Columbia, Canada; University of California, Berkeley

Moderator:

- **Justin Brookman,** Federal Trade Commission

PRIVACYCON

# BREAK

PRIVACYCON