HideMyApp: Hiding the Presence of Sensitive Apps on Android

Anh Pham\textsuperscript{1,2}, Italo Dacosta\textsuperscript{1}, Eleonora Losiouk\textsuperscript{3}, John Stephan\textsuperscript{1}, Kévin Huguenin\textsuperscript{4}, Jean-Pierre Hubaux\textsuperscript{1}

\textsuperscript{1}EPFL
\textsuperscript{2}ABB Switzerland
\textsuperscript{3}Uni. of Padova
\textsuperscript{4}Uni. of Lausanne
Mobile Health (mHealth)
Privacy Threat: Apps Fingerprinting Other Apps

The presence of an app already reveals sensitive information

Installed Apps:
- Diabetes
- Depression
- Cancer
- ...

Third-Party Servers
Research Questions

- Fingerprintability of apps
- Apps’ interest in fingerprinting other apps
- Our solution (HideMyApp)
Fingerprintability of Apps

Java API Framework
- w/o Permissions
- w/ Permissions

Linux-Layer Interface
- w/ Default Privilege
- w/ Debugging Privilege
Fingerprintability of Apps

Default privilege + No permissions
Fingerprintability of Apps

Default privilege + No permissions

Package name

- To retrieve the list of installed apps:
  - getInstalledApplications()
  - getInstalledPackages()

- To check if a specific app is installed:
  - getResourcesForApplication()
  - getPackageManager()
- ....

Removing methods or adding permissions is complicated.
Apps Inquiring about Other Apps

- Analysis on 2917 popular APKs from Google Play
- Static and dynamic analysis
- 19.2% to 57% of apps query for the list of installed apps
- Most requests come from third-party libs
- Free apps query for the list of installed apps more than paid apps

Apps want to fingerprint other apps and millions of users are affected.
Apps’ Compliance w/ Privacy Guidelines

From Google privacy guidelines:
- A list of installed apps (LIA) is sensitive
- Apps collecting LIA w/o users’ consent are classified as Mobile Unwanted Software

- From 2917 APKs, collected 2499 privacy policies
- Only 162 apps inform users about LIA collection
- 76 apps state that LIA is non-sensitive

Lack of effective protection mechanisms
Our Solution: HideMyApp (HMA)

- To host apps developed by the hospitals
- To (un)install and update apps
- To launch apps installed from the App Store

(App Store) controlled by hospitals
Adversarial Model

- Trusted and secure
- Wants to learn if a specific app is installed
  - Is nosy
    - Has default app privilege
    - Has all dangerous permissions

App Store
(controlled by hospitals)
HMA Overview

Request to retrieve an mHealth app

A container app

- Has a generic package name
- Obfuscated
  - Static information
  - Runtime information

App Store (controlled by hospitals)
Obfuscation: Static Information

- Generic package name
- Permissions
- Homogenized theme
- Randomized names for components
- Generic icon
- Generic label
- Resources loaded from the APK at runtime
Evaluation: Dataset

• 50 mHealth apps from Google Play

• Chosen based on their popularity, sensitivity and functionality

• Examples:

Beurer HealthManager  Cancer.Net Mobile  What's Up? - Mental Health
Evaluation Criteria and Implementation

• Implementation: [1]
  - HMA App Store
  - Manager App
  - Rely on DroidPlugin library for user-level virtualization [2]

[1] https://hma.epfl.ch
Cold-Start Delays: w/ and w/o HMA

Cold-start delays are less than 3 s
Conclusions

• Apps can and do fingerprint other apps
  - 57% of apps query for the list of apps

• Existing solutions are ineffective

• HMA: the first solution for hiding apps
  - Compatible with existing apps
  - Effective and usable
  - Runs on stock Android devices

https://hma.epfl.ch