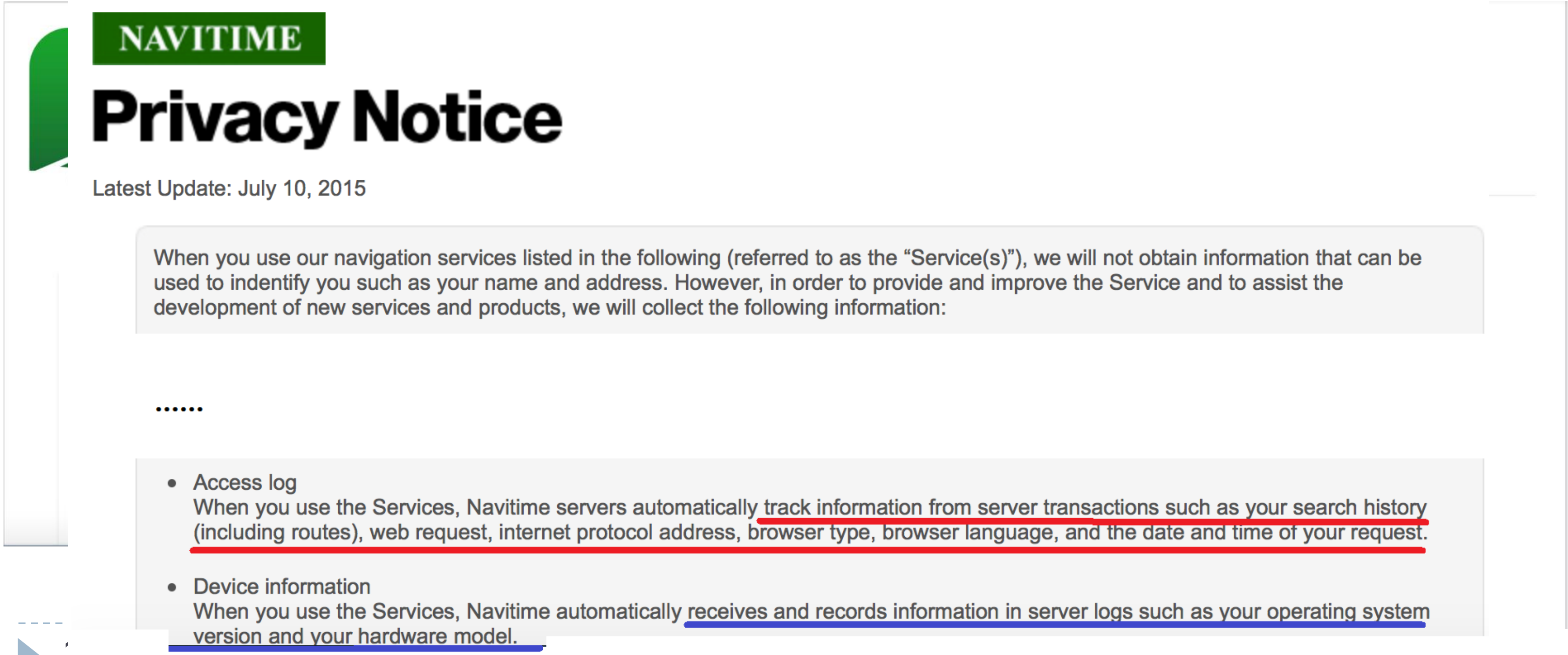
Would You Need Help to Create Privacy Policies for Apps?

Le Yu, Chenxiong Qian, Xiapu Luo, Lei Xue
The Hong Kong Polytechnic University

Privacy Policy

- Explain what data will be accessed/transmitted/stored/shared/used by the app as well as the reasons.
- Emphasize what data will not be accessed/transmitted/stored/shared/used by the app.



NAVITIME

Privacy Notice

Latest Update: July 10, 2015

When you use our navigation services listed in the following (referred to as the "Service(s)"), we will not obtain information that can be used to indentify you such as your name and address. However, in order to provide and improve the Service and to assist the development of new services and products, we will collect the following information:

……

- Access log
  When you use the Services, Navitime servers automatically track information from server transactions such as your search history (including routes), web request, internet protocol address, browser type, browser language, and the date and time of your request.
- Device information
  When you use the Services, Navitime automatically receives and records information in server logs such as your operating system version and your hardware model.

CNET › Internet

Path… sues

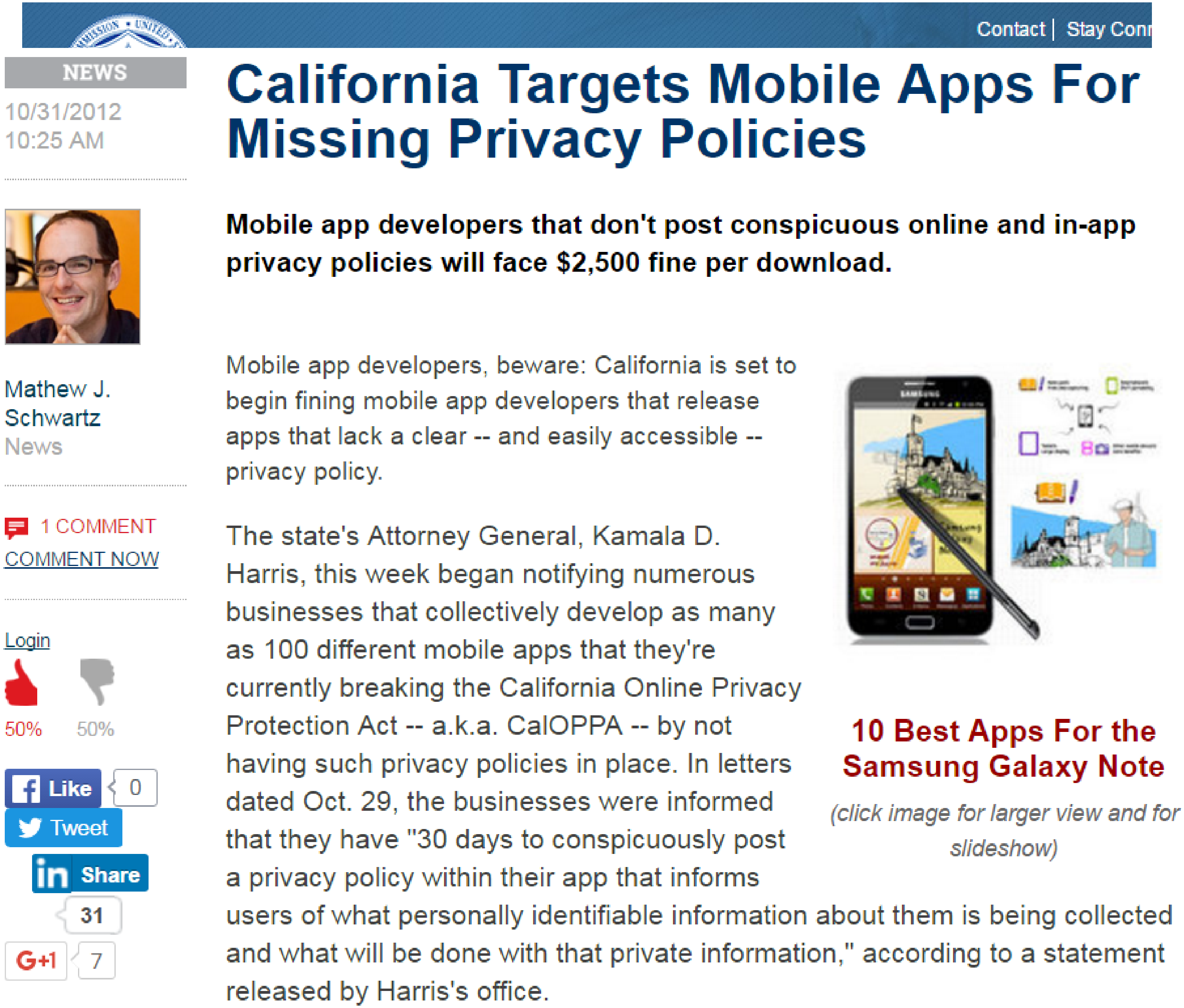
with…

The social-… …nt privacy assessmen…

Intern…

February …
9:20 AM

by Shara
@shara



Contact | Stay Conn…

NEWS

10/31/2012
10:25 AM

Mathew J. Schwartz
News

1 COMMENT
COMMENT NOW

Login

50% 50%

Like 0
Tweet
Share 31
G+1 7

California Targets Mobile Apps For Missing Privacy Policies

Mobile app developers that don't post conspicuous online and in-app privacy policies will face $2,500 fine per download.

Mobile app developers, beware: California is set to begin fining mobile app developers that release apps that lack a clear -- and easily accessible -- privacy policy.

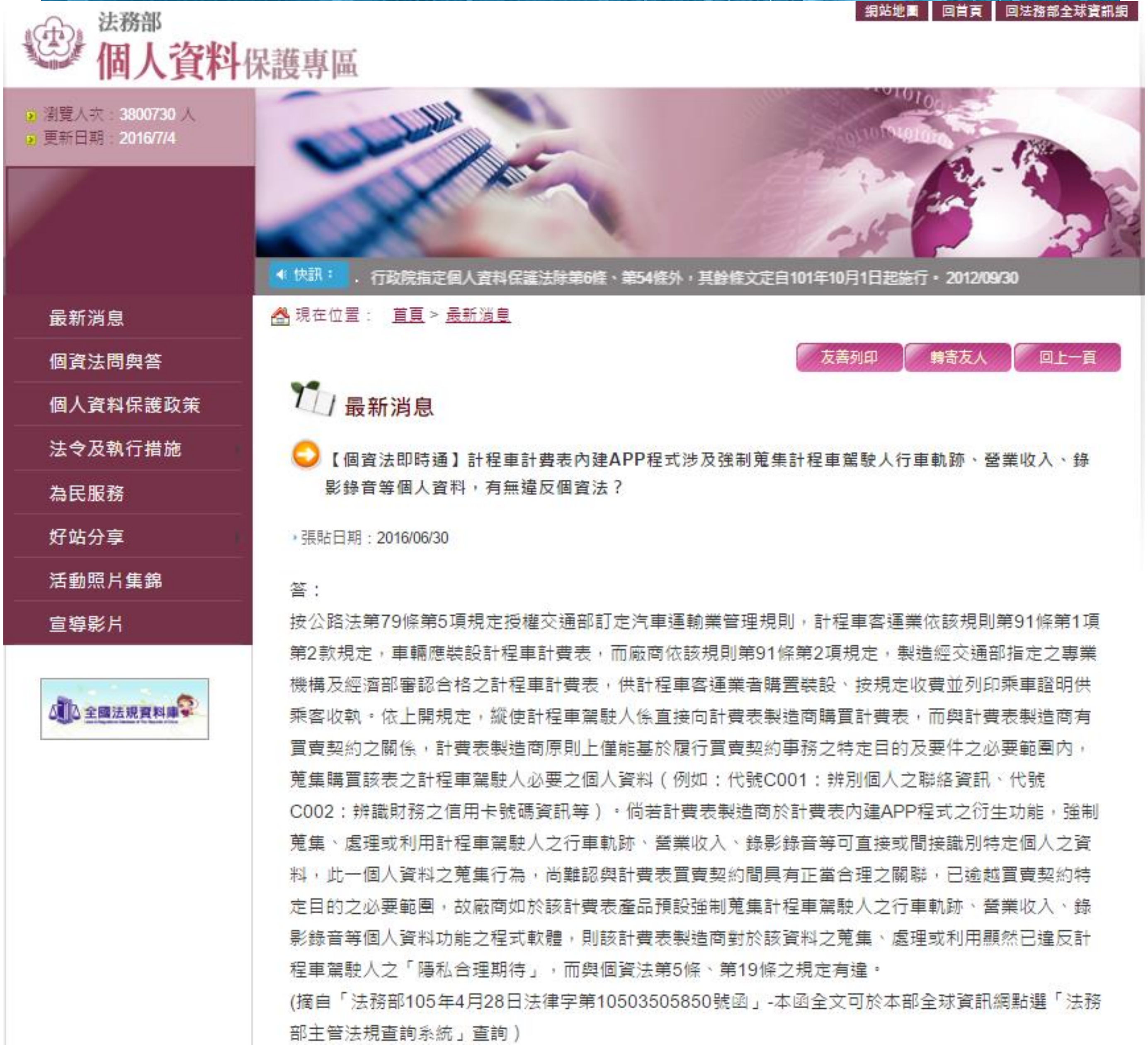
The state's Attorney General, Kamala D. Harris, this week began notifying numerous businesses that collectively develop as many as 100 different mobile apps that they're currently breaking the California Online Privacy Protection Act -- a.k.a. CalOPPA -- by not having such privacy policies in place. In letters dated Oct. 29, the businesses were informed that they have "30 days to conspicuously post a privacy policy within their app that informs users of what personally identifiable information about them is being collected and what will be done with that private information," according to a statement released by Harris's office.

10 Best Apps For the Samsung Galaxy Note

(click image for larger view and for slideshow)

information, even though it was shared automatically rendering the option meaningless.

PRIVACY & INFORMATION



最新消息

【個資法即時通】計程車計費表內建APP程式涉及強制蒐集計程車駕駛人行車軌跡、營業收入、錄影錄音等個人資料，有無違反個資法？

‧張貼日期：2016/06/30

答：

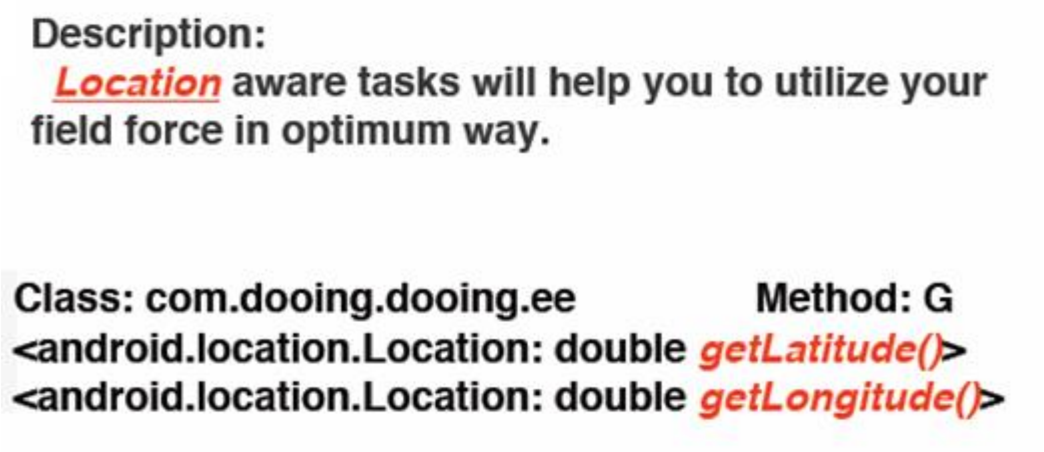
按公路法第79條第5項規定授權交通部訂定汽車運輸業管理規則，計程車客運業依該規則第91條第1項第2款規定，車輛應裝設計程車計費表，而廠商依該規則第91條第2項規定，製造經交通部指定之專業機構及經濟部審認合格之計程車計費表，供計程車客運業者購置裝設、按規定收費並列印乘車證明供乘客收執。依上開規定，縱使計程車駕駛人係直接向計費表製造商購買計費表，而與計費表製造商有買賣契約之關係，計費表製造商原則上僅能基於履行買賣契約事務之特定目的及要件之必要範圍內，蒐集購買該表之計程車駕駛人必要之個人資料（例如：代號C001：辨別個人之聯絡資訊、代號C002：辨識財務之信用卡號碼資訊等）。倘若計費表製造商於計費表內建APP程式之衍生功能，強制蒐集、處理或利用計程車駕駛人之行車軌跡、營業收入、錄影錄音等可直接或間接識別特定個人之資料，此一個人資料之蒐集行為，尚難認與計費表買賣契約間具有正當合理之關聯，已逾越買賣契約特定目的之必要範圍，故廠商如於該計費表產品預設強制蒐集計程車駕駛人之行車軌跡、營業收入、錄影錄音等個人資料功能之程式軟體，則該計費表製造商對於該資料之蒐集、處理或利用顯然已違反計程車駕駛人之「隱私合理期待」，而與個資法第5條、第19條之規定有違。
(摘自「法務部105年4月28日法律字第10503505850號函」-本函全文可於本部全球資訊網點選「法務部主管法規查詢系統」查詢）

Outline

- Spotting Issues in Apps' Privacy Policies
- Generating Privacy Policies Templates for Apps
- Conclusion

Common Faults in Privacy Policies

- **Incomplete privacy policy**
  - The privacy policy does not cover an app's all behaviors of accessing personal information.
  - Example
    - Get location information without claiming such behavior in its privacy policy.

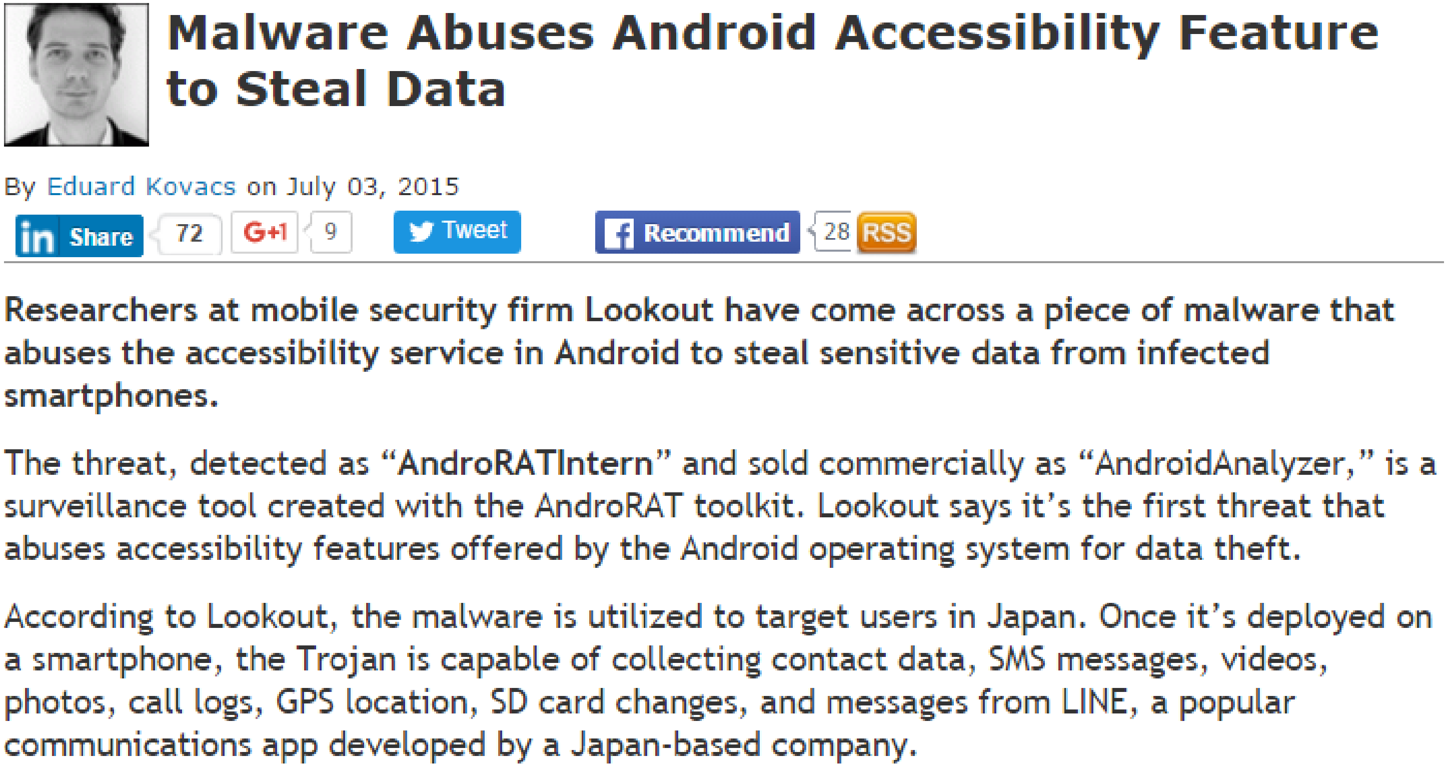


Description:
Location aware tasks will help you to utilize your field force in optimum way.

```
Class: com.dooing.dooing.ee                Method: G
<android.location.Location: double getLatitude()>
<android.location.Location: double getLongitude()>
```

Common Faults in Privacy Policies

- **Incorrect privacy policy**
  - An incorrect privacy policy declares that the app will not collect, use, retain, or disclose personal information, but the app does.
  - licy:



Malware Abuses Android Accessibility Feature to Steal Data

By Eduard Kovacs on July 03, 2015

Researchers at mobile security firm Lookout have come across a piece of malware that abuses the accessibility service in Android to steal sensitive data from infected smartphones.

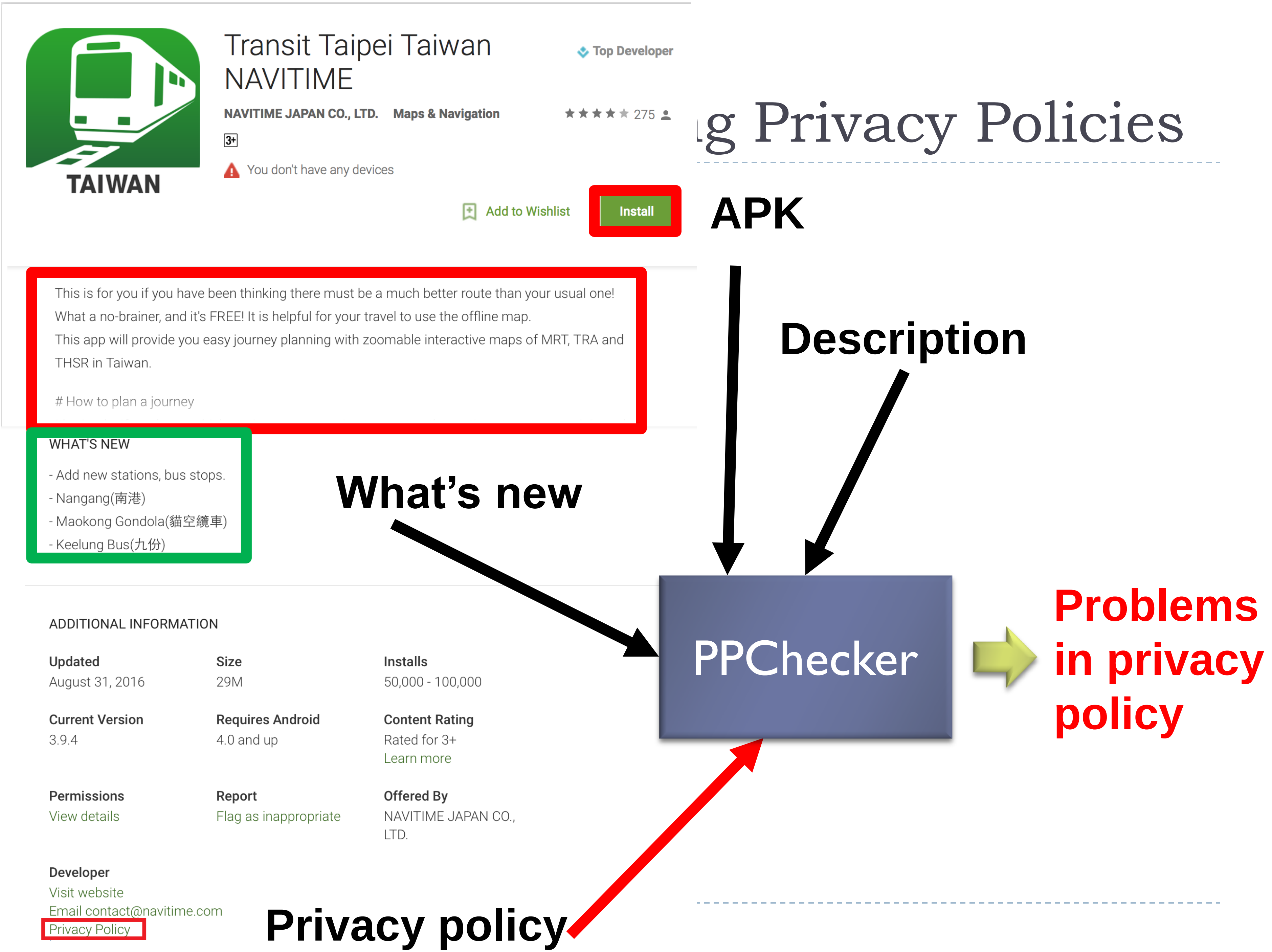
The threat, detected as “AndroRATIntern” and sold commercially as “AndroidAnalyzer,” is a surveillance tool created with the AndroRAT toolkit. Lookout says it’s the first threat that abuses accessibility features offered by the Android operating system for data theft.

According to Lookout, the malware is utilized to target users in Japan. Once it’s deployed on a smartphone, the Trojan is capable of collecting contact data, SMS messages, videos, photos, call logs, GPS location, SD card changes, and messages from LINE, a popular communications app developed by a Japan-based company.

Common Faults in Privacy Policies

- **Inconsistent privacy policy**
  - The privacy policy of an app is in conflict with that of its third-party libs.
  - Example
    - A popular game app's privacy policy says:
      - □ *"we do not use or collect your precise geographic location."*
    - It uses a third-party library, whose privacy policy says:
      - □ *"We receive information about Users, their devices, locations and interactions with the Service primarily in two ways."*

g Privacy Policies



Transit Taipei Taiwan NAVITIME

Top Developer

NAVITIME JAPAN CO., LTD. Maps & Navigation

★★★★★ 275

3+

TAIWAN

You don't have any devices

Add to Wishlist

Install

This is for you if you have been thinking there must be a much better route than your usual one!
What a no-brainer, and it's FREE! It is helpful for your travel to use the offline map.
This app will provide you easy journey planning with zoomable interactive maps of MRT, TRA and THSR in Taiwan.

How to plan a journey

WHAT'S NEW

- Add new stations, bus stops.
- Nangang(南港)
- Maokong Gondola(貓空纜車)
- Keelung Bus(九份)

ADDITIONAL INFORMATION

Updated	Size	Installs
August 31, 2016	29M	50,000 - 100,000
Current Version	**Requires Android**	**Content Rating**
3.9.4	4.0 and up	Rated for 3+ Learn more
Permissions	**Report**	**Offered By**
View details	Flag as inappropriate	NAVITIME JAPAN CO., LTD.

Developer

Visit website

Email contact@navitime.com

Privacy Policy

APK

Description

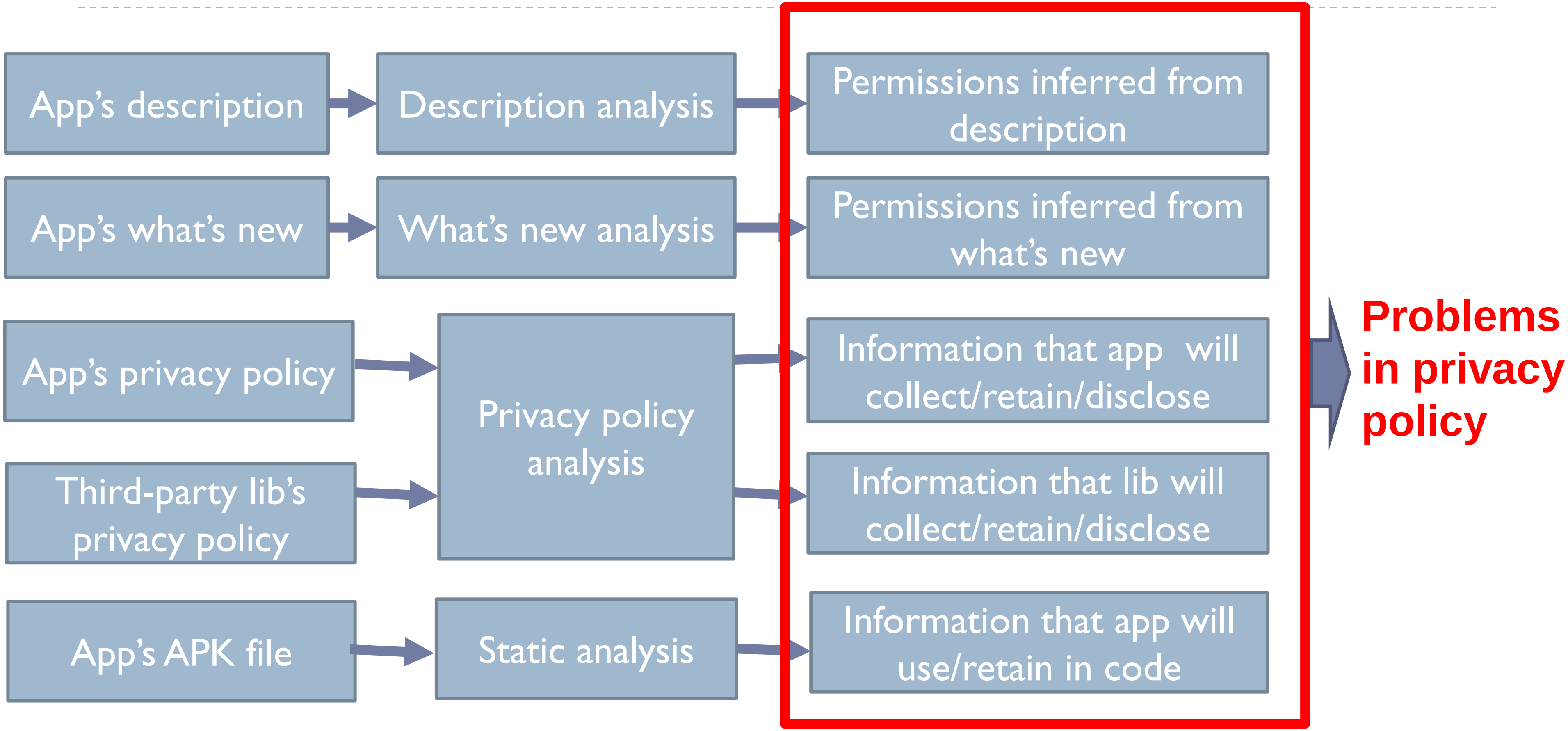
What’s new

Privacy policy

PPChecker

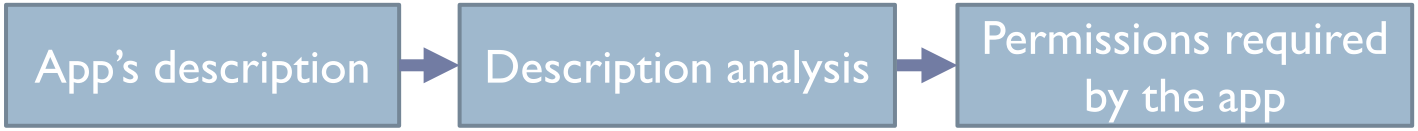
Problems in privacy policy

PPChecker


App's description
Description analysis
Permissions inferred from description
App's what's new
What's new analysis
Permissions inferred from what's new
App's privacy policy
Third-party lib's privacy policy
Privacy policy analysis
Information that app will collect/retain/disclose
Information that lib will collect/retain/disclose
App's APK file
Static analysis
Information that app will use/retain in code
Problems in privacy policy

Our tool analyzed 1,680 popular apps downloaded from Google play store and found that 484 apps (i.e., **28.8%**) contain at least one kind of problem.

Description Analysis


App's description → Description analysis → Permissions required by the app

- Example:
  - “*share your location*” ➔ ACCESEE_FINE_LOCATION
  - “*exchange contacts*” ➔ READ_CONTACTS

- 11 Permissions
  - WRITE_EXTERNAL_STORAGE
  - ACCESS_FINE_LOCATION
  - ACCESS_COARSE_LOCATION
  - GET_ACCOUNTS
  - RECEIVE_BOOT_COMPLETED
  - READ_CALENDAR

  CAMERA
  READ_CONTACTS
  RECORD_AUDIO
  WRITE_SETTINGS
  WRITE_CONTACTS

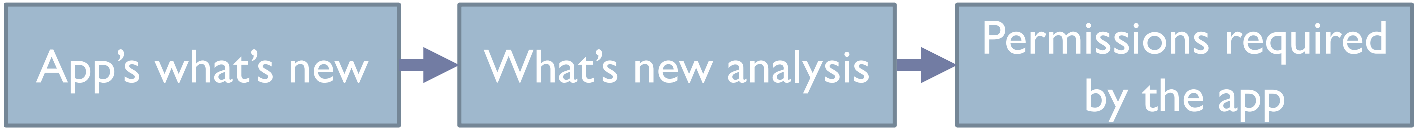
Description Analysis

- AutoCog:
  - Generate a description-to-permission relatedness (DPR).
    - ACCESS_FINE_LOCATION ←→(“*view*”, “*map*”), (“*search*”, “*parking*”)
  - Extract (verb, noun) pairs from sentences in description.
    - “*share your location*” → (“*share*”, “*location*”)
  - Calculate the semantic similarity between (verb, noun) pair and the DPR model.
    - (“*search*”, “*hotel*”) ←→ (“*find*”, “*hotel*”) (ACCESS_FINE_LOCATION)

Description Analysis

- Enhanced AutoCog:
  - Remove *Negative* sentences in description.
    - “you do *not* need to create another contact list …”
  - Co-reference resolution: Pronouns in description.
    - Before: “*and meet new friends in person, bringing them from* …”
    - After: “*and meet new friends in person, bringing new friend from* …”

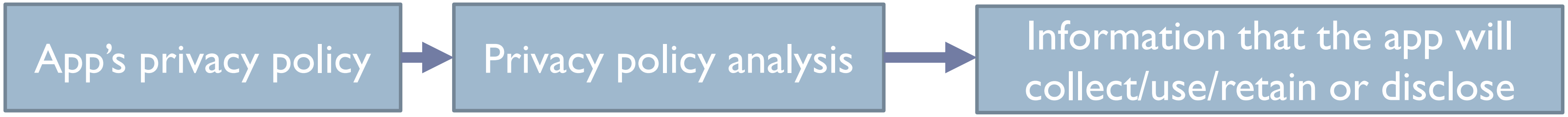
What's new Analysis



- Use enhanced AutoCog to process the what's new
- 11 Permissions
  - WRITE_EXTERNAL_STORAGE
  - ACCESS_FINE_LOCATION
  - ACCESS_COARSE_LOCATION
  - GET_ACCOUNTS
  - RECEIVE_BOOT_COMPLETED
  - READ_CALENDAR

  CAMERA
  READ_CONTACTS
  RECORD_AUDIO
  WRITE_SETTINGS
  WRITE_CONTACTS

Privacy Policy Analysis

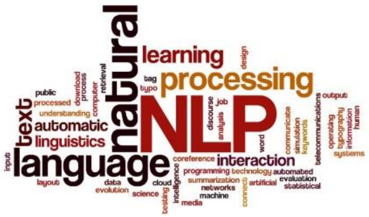


App's privacy policy → Privacy policy analysis → Information that the app will collect/use/retain or disclose

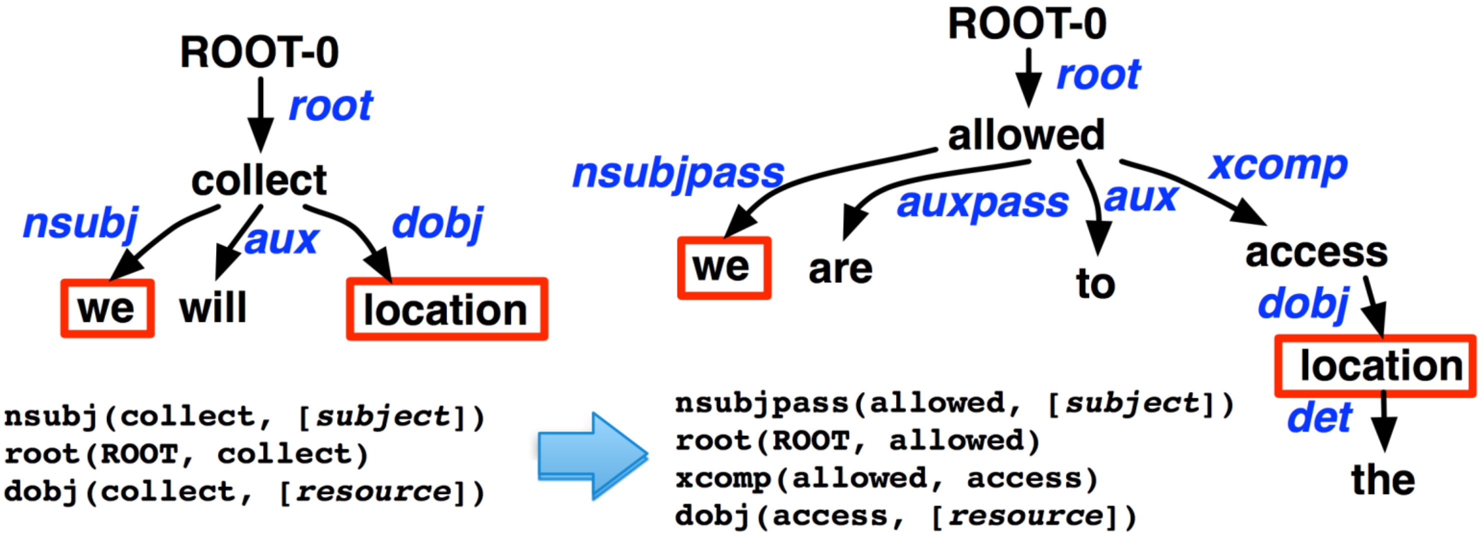
- How to handle different sentences representing the same meaning?
  - "*we will collect your location*".
  - "*your location information is collected by the app*".

- Key Idea: Summarize the semantic patterns (i.e., sentence structure) used in privacy policy.

Privacy Policy Analysis



- Bootstrap algorithm: Automatically find pattern in corpus
  - Step 1: Seed pattern:
    - “[*sbj*] collect/use/retain/disclose [*resource*]”.
  - Step 2: Find the sentences that contain the same subject, resource.
  - Step 3: Extract pattern from new sentences.



```
nsubj(collect, [subject])
root(ROOT, collect)
dobj(collect, [resource])
```

```
nsubjpass(allowed, [subject])
root(ROOT, allowed)
xcomp(allowed, access)
dobj(access, [resource])
```

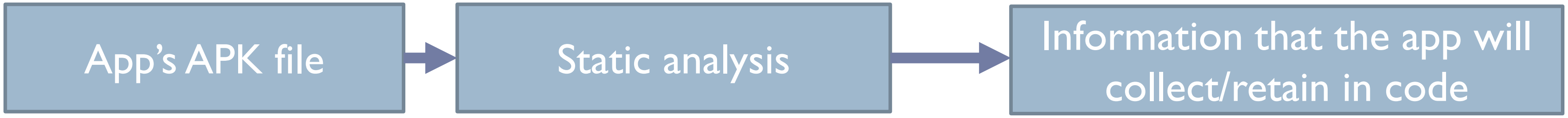
Privacy Policy Analysis

- Pattern match: Find sentences that match the patterns.
  - Pattern: [*sbj*] "*be allowed to*" VP_{collect} [*resource*].
  - Sentence: "*We are allowed to collect your location*".
- Negation analysis: Identify negative sentences.
  - Information will not be collected/used/retained.
  - Method:
    - Maintain a list of negation words.
    - Check subject: "*No information will be collected*".
    - Check root-word related words: "*We will not collect your location*".

Privacy Policy Analysis

- Constraint analysis:
  - Under which condition the information will be collected/retained.
  - Example:
    - Category: Registration.
    - Extract pre-condition/post-condition: "*When you register with or visit the rockyou sites*".
    - Search keywords: "register".
- Extract resource from sentence.
  - Example:
    - "Your *location* will be collected by the app"
    - Extract "*location*"

Static Analysis



App's APK file → Static analysis → Information that the app will collect/retain in code

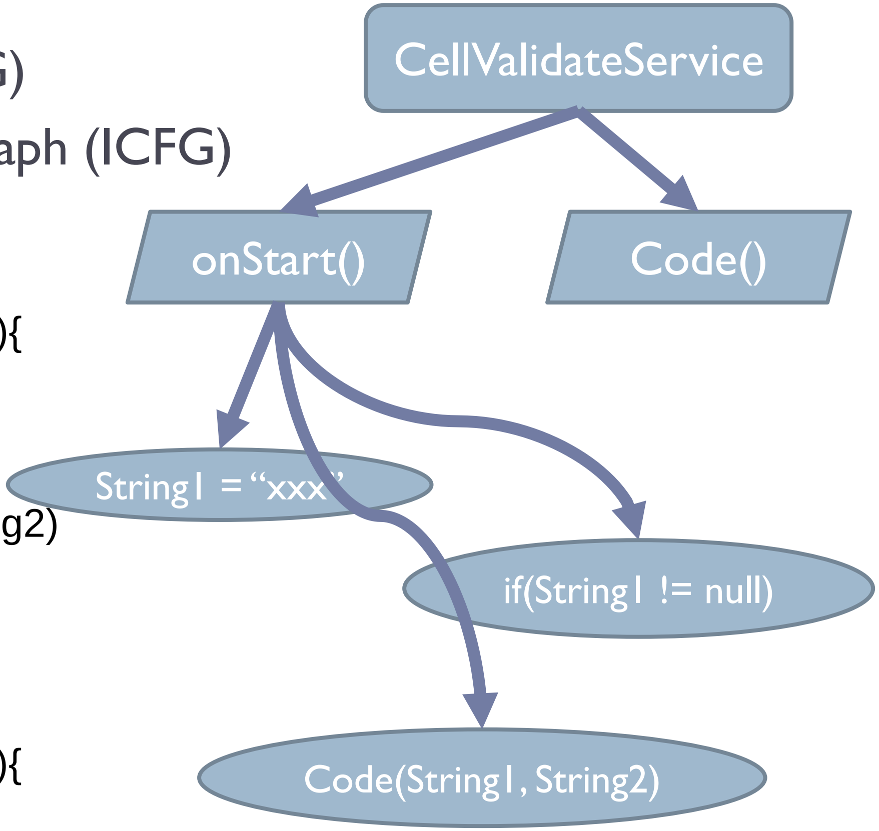
- Determine the collected information.
  - Check APIs called in code. getLastKnownLocation() ➔ "location"
  - Check URIs used in code. content://com.android.contacts ➔ "contact"
- Determine the retained information.
  - Static taint analysis
    - □ Traverse the data dependency graph (DDG)
    - □ Source to sink path. getLatitude() ➔ sendTextMessage()

Static Analysis

- Implementation
  - Pre-processing
    - Extract dex file from APK file.
    - Transform dex file into intermediate language Shimple using soot.
  - Construct the Android Property Graph:
    - Abstract syntactic trees (ASTs)
    - Method call graph (MCG)
    - System dependency graph (SDG)
    - Inter-procedure control flow graph (ICFG)

Static Analysis

- Android Property Graph (APG):
  - Abstract syntactic trees (ASTs)
  - Method call graph (MCG)
  - System dependency graph (SDG)
  - Inter-procedure control flow graph (ICFG)



CellValidateService

onStart()

Code()

String1 = "xxx"

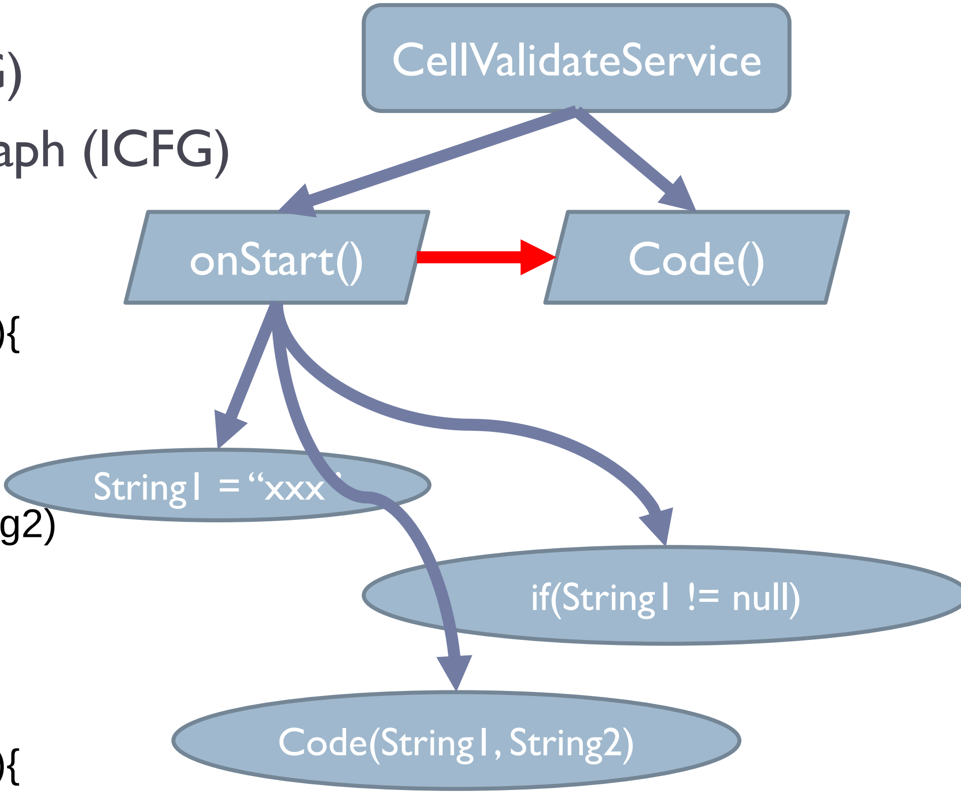
if(String1 != null)

Code(String1, String2)

```
public class  CellValidateService extends Service {
            public void onStart(Intent pIntent, int pInt){
                        String String1 = “xxx”
                        if(String1 != null)
                        {
                                    Code(String1, String2)
                                    …
                        }
            }

            private void Code(String dest, String text){
            }
}
```

Static Analysis

- Android Property Graph:
  - Abstract syntactic trees (ASTs)
  - Method call graph (MCG)
  - System dependency graph (SDG)
  - Inter-procedure control flow graph (ICFG)

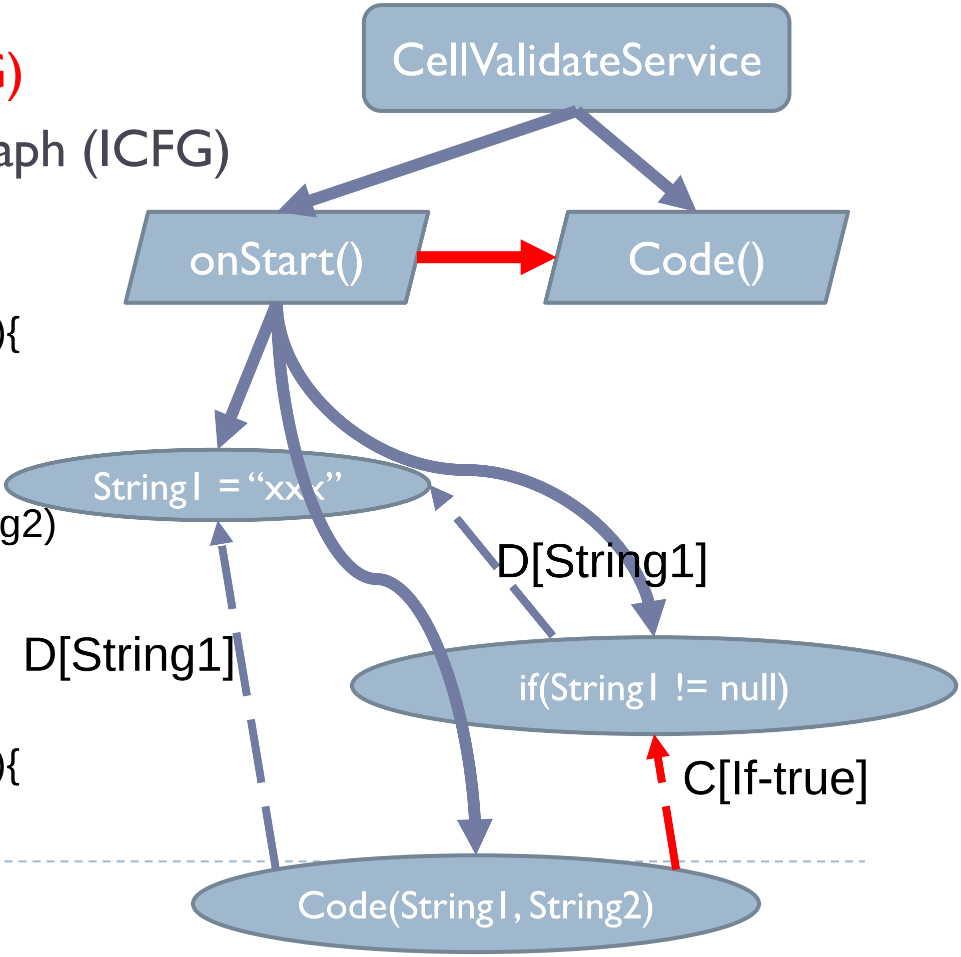


```
public class  CellValidateService extends Service {
            public void onStart(Intent pIntent, int pInt){
                        String String1 = “xxx”
                        if(String1 != null)
                        {
                                    Code(String1, String2)
                                    …
                        }
            }

            private void Code(String dest, String text){
            }
}
```

Static Analysis

- Android Property Graph:
  - Abstract syntactic trees (ASTs)
  - Method call graph (MCG)
  - System dependency graph (SDG)
  - Inter-procedure control flow graph (ICFG)

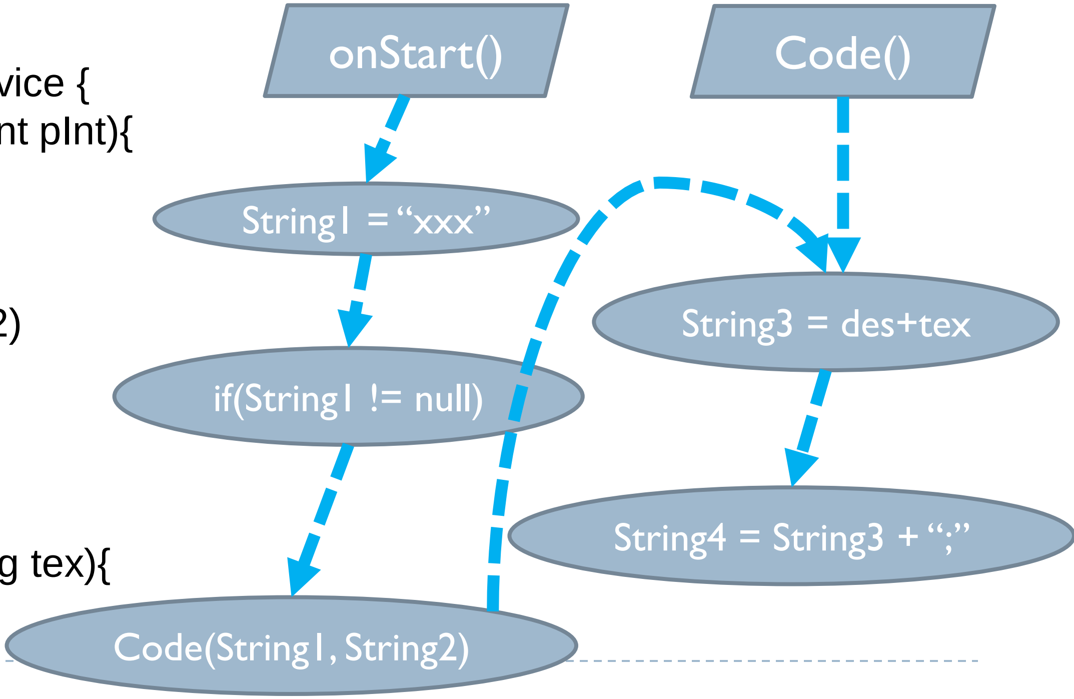
```
public class  CellValidateService extends Service {
            public void onStart(Intent pIntent, int pInt){
                        String String1 = “xxx”
                        if(String1 != null)
                        {
                                    Code(String1, String2)
                                    …
                        }
            }

            private void Code(String dest, String text){
            }
}
```



Static Analysis

- Android Property Graph:
  - Abstract syntactic trees (ASTs)
  - Method call graph (MCG)
  - System dependency graph (SDG)
  - Inter-procedure control flow graph (ICFG)

```
public class  CellValidateService extends Service {
        public void onStart(Intent pIntent, int pInt){
                String String1 = “xxx”
                if(String1 != null)
                {
                    Code(String1, String2)
                    …
                }
        }

        private void Code(String des, String tex){
                String3 = des + tex
                String4 = String3 + “;”
        }
}
```



Problem Identification

- Incomplete privacy policy
  - Contrast description with privacy policy
    - The permissions inferred from description vs. the permissions associated with the information mentioned in privacy policy.
  - Contrast what'snew with privacy policy
    - The permissions inferred from what'snew vs. the permissions associated with the information mentioned in privacy policy.
  - Contrast code with privacy policy
    - The information collected/retained by code vs. the information mentioned in privacy policy.

Problem Identification

- Incorrect privacy policy
  - Contrast description with privacy policy
    - Privacy policy declares *NOT* to use the information whose permissions can be inferred from description.
  - Compare code with privacy policy
    - Privacy policy declares *NOT* to use the information that is collected/retained by code.

Problem Identification

- Inconsistent privacy policy

  - Compare an app's privacy policy with its third-party libs' privacy policies
    - App's privacy policy declares *NOT* to collect/use/retain/disclose certain information.
    - Lib privacy policy declares to collect/use/retain/disclose the information

Experimental Result

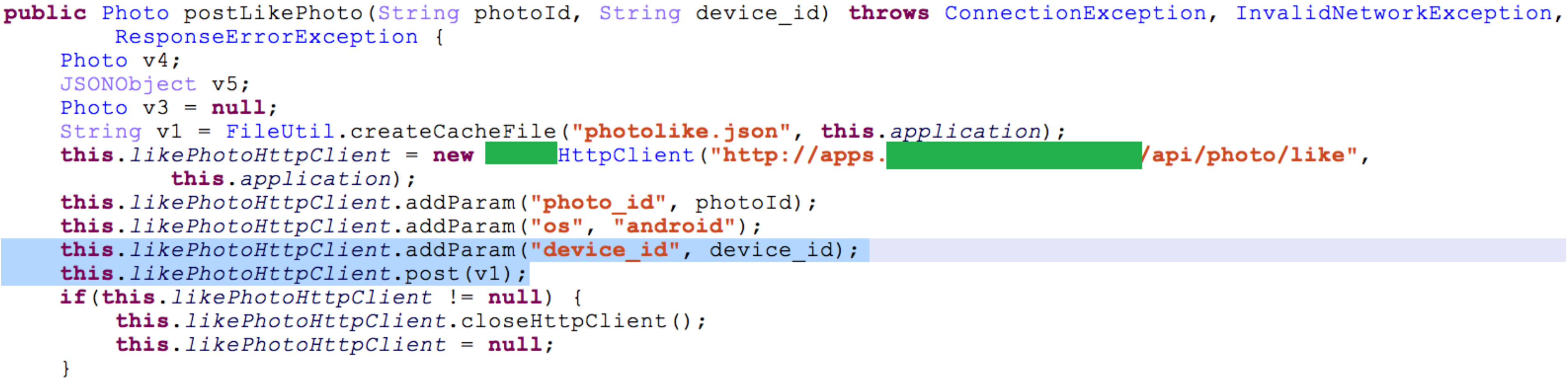
- Data set:
  - 1680 apps downloaded from Google Play
    - Each app contains APK file, description, what’snew, and privacy policy
    - 484 apps (i.e., **28.8%**) contain at least one kind of problem.
      - Some apps have more than one problem.

  - Third-party libraries considered:
    - Contain privacy policies and class names.
    - 52 ad libs. Example: Admob, Tapjoy
    - 9 social libs. Example: Facebook, Twitter
    - 20 development tools. Example: Flurry analytics, Unity 3D

Incomplete Privacy Policy

- 392 questionable apps

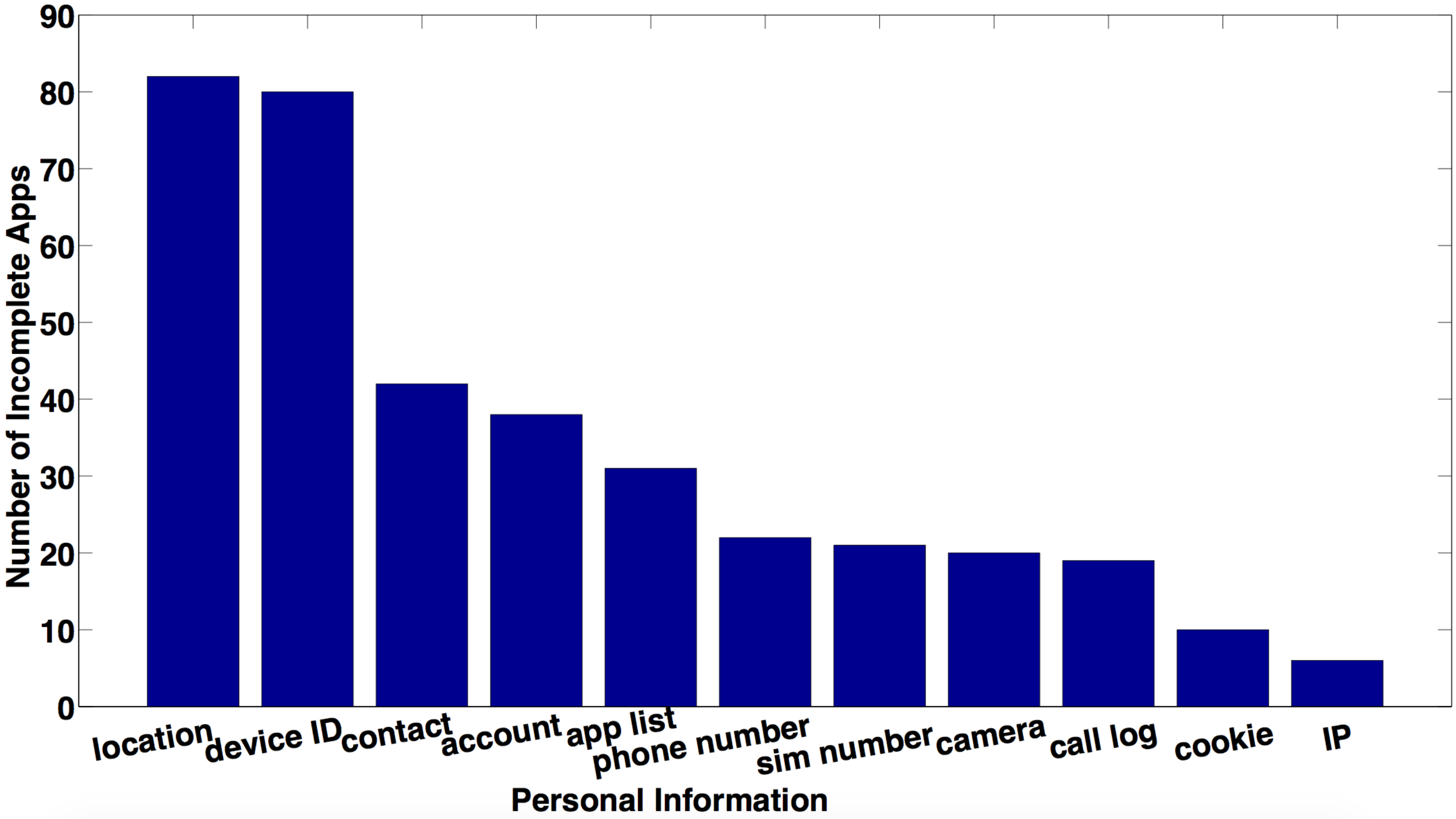
- Example
  - Privacy policy
    - "Internet technology requires some basic information in order for users of websites or mobile apps to use our online services smoothly. The basic information should include but not limited to IP addresses and/or domain names, browser type and settings, language settings, geographical district, operating system, and time/duration of visit. These data are anonymous and cannot be used to identify the user under general situations. *When personal identifiable information of users are involved or collected in the Federation's websites or mobile apps, users will be prompted to give explicit alert of the collection so as to give consent*."
  - Code:



```
public Photo postLikePhoto(String photoId, String device_id) throws ConnectionException, InvalidNetworkException,
    ResponseErrorException {
  Photo v4;
  JSONObject v5;
  Photo v3 = null;
  String v1 = FileUtil.createCacheFile("photolike.json", this.application);
  this.likePhotoHttpClient = new [redacted]HttpClient("http://apps.[redacted]/api/photo/like",
      this.application);
  this.likePhotoHttpClient.addParam("photo_id", photoId);
  this.likePhotoHttpClient.addParam("os", "android");
  this.likePhotoHttpClient.addParam("device_id", device_id);
  this.likePhotoHttpClient.post(v1);
  if(this.likePhotoHttpClient != null) {
    this.likePhotoHttpClient.closeHttpClient();
    this.likePhotoHttpClient = null;
  }
```

Incomplete Privacy Policy

- Number of incomplete privacy policies for different kinds of personal information.



Incorrect Privacy Policy

- 4 questionable apps.

- Example 1
  - Privacy policy
    - *"we are not collecting your data of birth, phone number, name or other personal information, nor those of your contact."*
  - Code
    - Collect contact information.

- Example 2
  - Ambiguous privacy policy
    - *"we will not store your real phone number, name and contacts"*
    - *"Users locations would not be transmitted out from the app".*
  - Code
    - They get the information and write to log file.
    - Note: third-party libs in the app or colluded apps can access the information in log file; attackers could use adb to access the log; apps in a rooted smartphone could access the log.

Inconsistent Privacy Policy

- 111 questionable apps

- Example
  - App's privacy policy
    - *"we do not collect information such as your real name, address, or phone number."*
  - Third-party lib's privacy policy
    - *"we may collect device specific information (such as ... mobile network information including phone number)".*

- Example
  - App's privacy policy
    - *"We don't share your personal information with any third parties".*
  - Third-party lib's privacy policy
    - *"We may share certain types of personal information with third parties."*

Checking 21 popular apps in Taiwan

- Three incomplete privacy policies
  - One app missed installed app list.
  - One app missed location.
  - One app missed account.
- One inconsistent privacy policy
  - One app declares "*we do not share personal information with third party advertisers for their direct marketing purpose*".
  - Third party lib declares "XXX may also share your personal information with third parties"

Problems in Third-Party Libs’ Privacy Policies

- 52 ad libs, 9 social libs, and 20 development tools

- 5 third-party libs have incomplete privacy policies.
  - 3 libs missed device ID
  - 2 libs missed IP address
  - 1 lib missed location
  - 1 lib missed sim card number

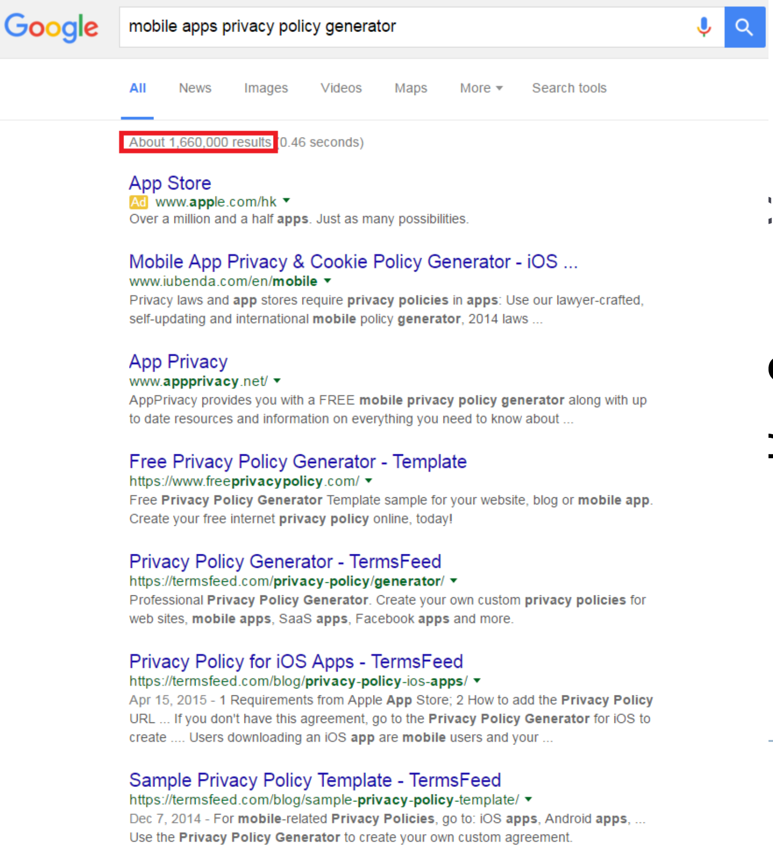
- Example
  - A lib will collect location information according to its code.
    - getLatitude()
    - getLongitude()
  - However, its privacy policy does not mention such behavior.

Outline

- Spotting Issues in Apps' Privacy Policies
- Generating Privacy Policies Templates for Apps
- Conclusion

Writing Privacy Policy

- Is it difficult?
  - **No**, because
    - There are many guidelines and training courses.
    - Free online generators.
    - …
  - **Yes**, becau … y may
    - not well u
    - not be far … each API used.
    - not know … d-party libraries.
    - …



About 1,660,000 results (0.46 seconds)

App Store
Ad www.apple.com/hk ▾
Over a million and a half **apps**. Just as many possibilities.

Mobile App Privacy & Cookie Policy Generator - iOS ...
www.iubenda.com/en/**mobile** ▾
Privacy laws and **app** stores require **privacy policies** in **apps**: Use our lawyer-crafted, self-updating and international **mobile** policy **generator**, 2014 laws ...

App Privacy
www.**appprivacy**.net/ ▾
AppPrivacy provides you with a FREE **mobile privacy policy generator** along with up to date resources and information on everything you need to know about ...

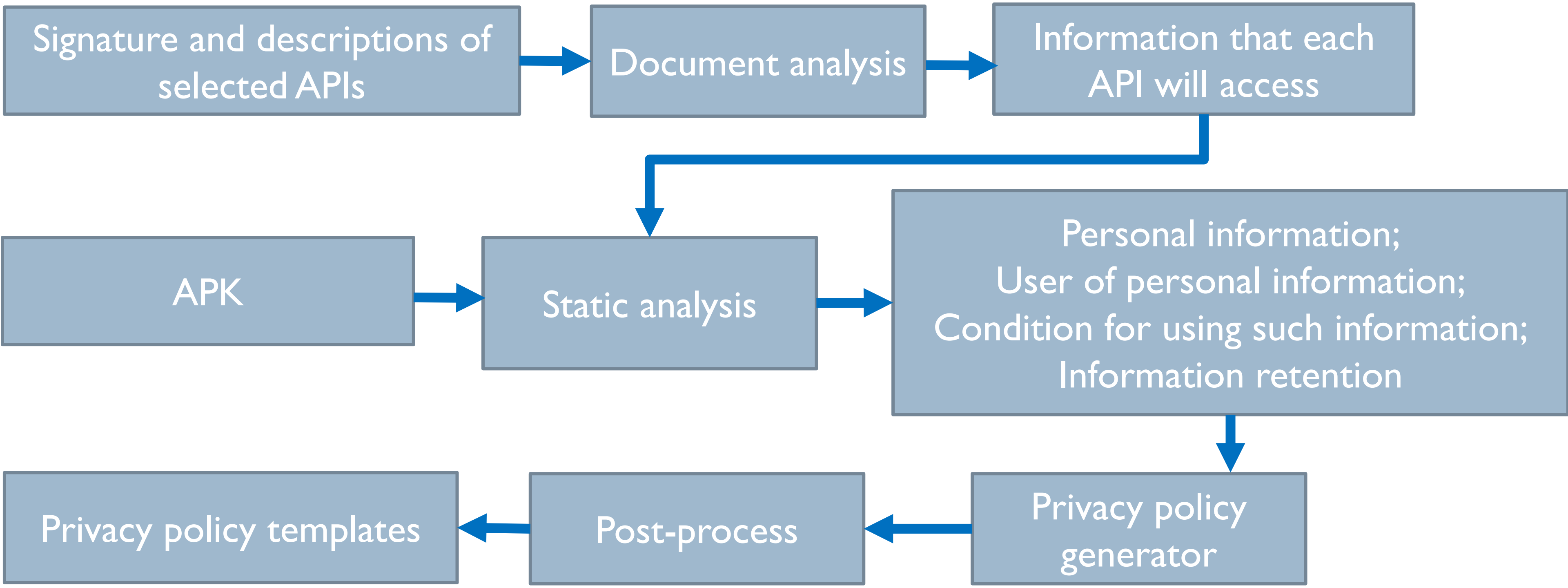
Free Privacy Policy Generator - Template
https://www.free**privacypolicy**.com/ ▾
Free **Privacy Policy Generator** Template sample for your website, blog or **mobile app**. Create your free internet **privacy policy** online, today!

Privacy Policy Generator - TermsFeed
https://termsfeed.com/**privacy-policy**/**generator**/ ▾
Professional **Privacy Policy Generator**. Create your own custom **privacy policies** for web sites, **mobile apps**, SaaS **apps**, Facebook **apps** and more.

Privacy Policy for iOS Apps - TermsFeed
https://termsfeed.com/blog/**privacy-policy**-ios-**apps**/ ▾
Apr 15, 2015 - 1 Requirements from Apple App Store; 2 How to add the **Privacy Policy** URL ... If you don't have this agreement, go to the **Privacy Policy Generator** for iOS to create .... Users downloading an iOS **app** are **mobile** users and your ...

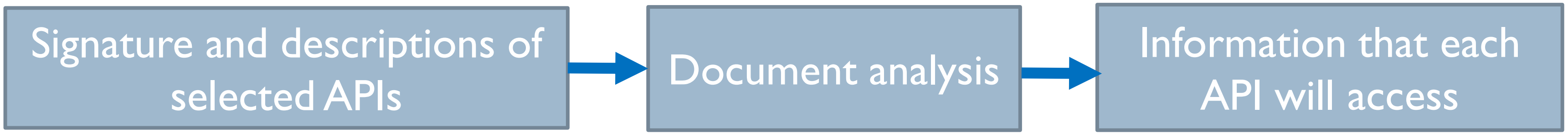
Sample Privacy Policy Template - TermsFeed
https://termsfeed.com/blog/sample-**privacy-policy**-template/ ▾
Dec 7, 2014 - For **mobile**-related **Privacy Policies**, go to: iOS **apps**, Android **apps**, ...
Use the **Privacy Policy Generator** to create your own custom agreement.

AutoPPG: Automatically Generating Privacy Policy Templates


Signature and descriptions of selected APIs
Document analysis
Information that each API will access
APK
Static analysis
Personal information;
User of personal information;
Condition for using such information;
Information retention
Privacy policy generator
Post-process
Privacy policy templates

Our tool takes in an apk file and then generate a correct and readable privacy policy template for it.

Document Analysis



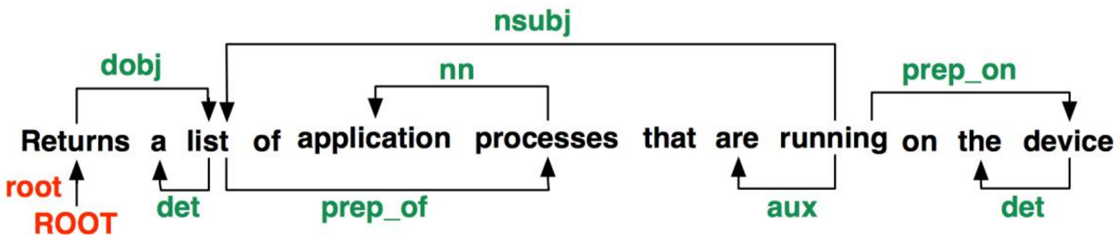
Signature and descriptions of selected APIs

Document analysis

Information that each API will access

- Syntactic analysis on the description of the API.
  - *getRunningAppProcesses(): "Returns a list of application processes that are running on the device".*

```
(ROOT
  (S
    (VP (VB Returns)
      (NP
        (NP (DT a) (NN list))
        (PP (IN of)
          (NP (NN application) (NNS processes)))
        (SBAR
          (WHNP (WDT that))
          (S
            (VP (VBP are)
              (VP (VBG running)
                (PP (IN on)
                  (NP (DT the) (NN device))))))))))
```

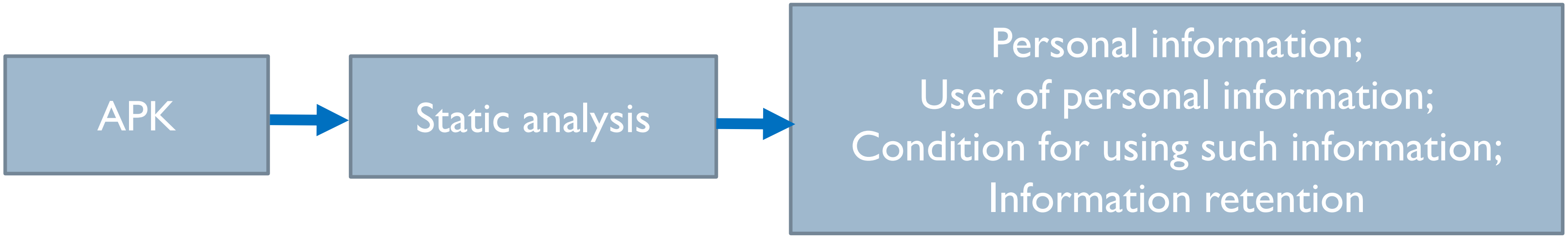

nsubj
dobj
nn
prep_on
Returns a list of application processes that are running on the device
root
ROOT
det
prep_of
aux
det

Document Analysis

- Extract noun phrase from method name.
  - *getRunningAppProcesses* ➔ Running App Process
- Extract noun phrase from class name.
  - Android.hardware.Camera ➔ Camera
- Extract private information.
  - Compare the object in description with method name/class name.
  - If the object cannot cover method name/class name, add additional information.
    - □ "list" "Running App Process" ➔ Low similarity
    - □ "list" + "of application processes"

Static Analysis

public
Integer
next() {
for(int i = p.length - 1; i >= 0;
(++p[i]
else
return p;
}
throw new NoSuchElementException();


APK
Static analysis
Personal information;
User of personal information;
Condition for using such information;
Information retention

- Extract Dex file from APK file (Unpacking if need)

- Construct Android Property Graph:
  - method call graph (MCG)
  - system dependency graph (SDG)
  - Inter-procedure control flow graph (ICFG)
  - Abstract syntactic trees (ASTs)

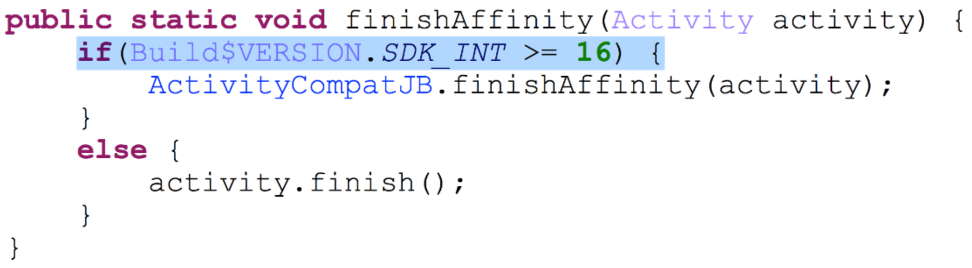
Static Analysis

- Identify the APIs/URIs used in code.
  - Get information through API
  - Get information through URI

- Reachability analysis to remove infeasible code.
  - Infeasible code will not be triggered.
  - Traversal the method call graph (MCG).
    - Two kinds of entry points
      - □ Lifecycle methods: onCreate()
      - □ UI callbacks: onClick()

Static Analysis

- Identify the conditions under which these APIs/URIs are used.
  - Device specific information.
    - Language: Locale.getDisplayLanguage()
    - OS version: android.os.Build.VERSION
    - Screen size: Display.getSize()
  - Natural environment requirement.
    - Time: Date.getHour()
    - Location: Location.getLatitude()



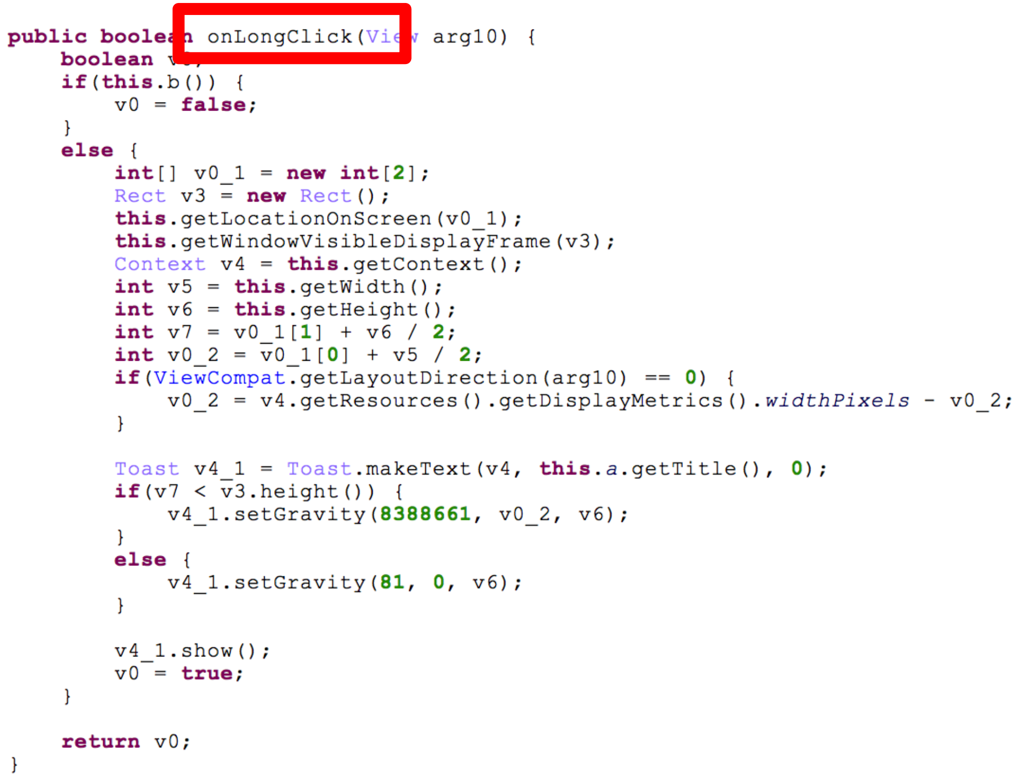
```
public static void finishAffinity(Activity activity) {
    if(Build$VERSION.SDK_INT >= 16) {
        ActivityCompatJB.finishAffinity(activity);
    }
    else {
        activity.finish();
    }
}
```

  - Hardware events.
    - Press BACK and HOME keys
    - Lifecycle callback: onPause(), onResume()

Static Analysis

- Identify the conditions under which these APIs/URIs are used.

  - UI events.
    - Widget: View.Button
    - Callback: onClick()
  - System events.
    - Broadcast receiver
    - Intent: BOOT_COMPLETED
  - Device status.
    - Current status of current device
    - API: PowerManager.isScreenOn()

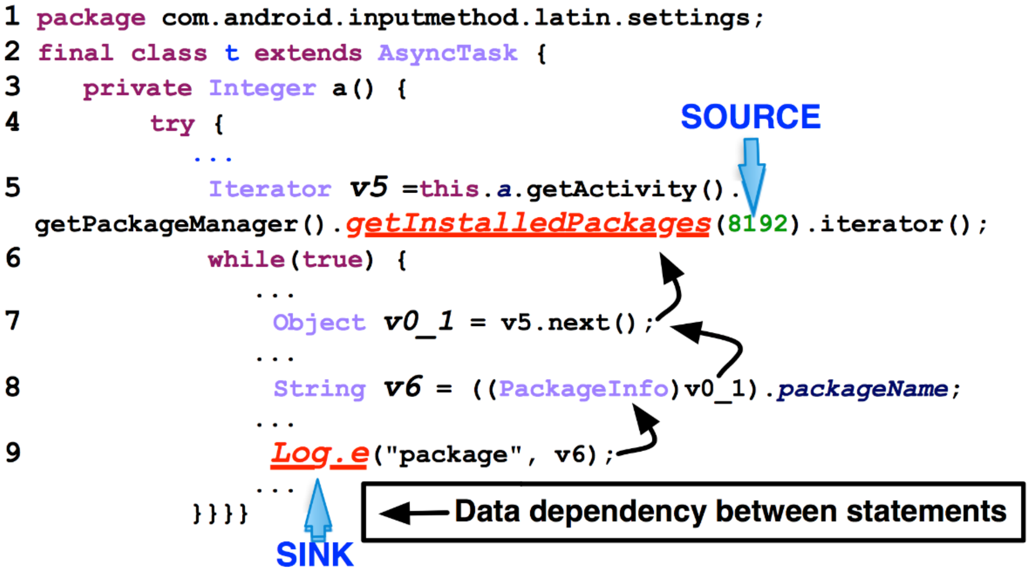


```
public boolean onLongClick(View arg10) {
    boolean v0;
    if(this.b()) {
        v0 = false;
    }
    else {
        int[] v0_1 = new int[2];
        Rect v3 = new Rect();
        this.getLocationOnScreen(v0_1);
        this.getWindowVisibleDisplayFrame(v3);
        Context v4 = this.getContext();
        int v5 = this.getWidth();
        int v6 = this.getHeight();
        int v7 = v0_1[1] + v6 / 2;
        int v0_2 = v0_1[0] + v5 / 2;
        if(ViewCompat.getLayoutDirection(arg10) == 0) {
            v0_2 = v4.getResources().getDisplayMetrics().widthPixels - v0_2;
        }

        Toast v4_1 = Toast.makeText(v4, this.a.getTitle(), 0);
        if(v7 < v3.height()) {
            v4_1.setGravity(8388661, v0_2, v6);
        }
        else {
            v4_1.setGravity(81, 0, v6);
        }

        v4_1.show();
        v0 = true;
    }

    return v0;
}
```

Static Analysis

- Identify the user of these APIs/URIs.
  - The app itself or third-party lib.

- Check if the information is stored in file/log, sent out through internet/SMS.
  - Static taint analysis on data dependency graph (DDG).
  - Source to sink path.

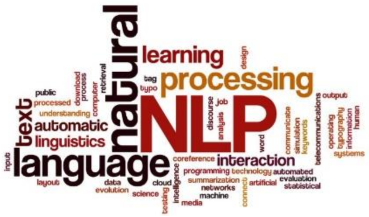


```
package com.android.inputmethod.latin.settings;
final class t extends AsyncTask {
   private Integer a() {
        try {
          ...
           Iterator v5 =this.a.getActivity().
getPackageManager().getInstalledPackages(8192).iterator();
           while(true) {
              ...
              Object v0_1 = v5.next();
              ...
              String v6 = ((PackageInfo)v0_1).packageName;
              ...
              Log.e("package", v6);
              ...
          }}}}
```

Privacy Policy Generator

- Template of each generated sentence:
  - *Sentence* = [pre-condition] *subject verb object* [post-*condition*]
  - Subject: User of the sensitive information.
  - Verb: Analyse the data flow to determine verb.
    - URIs:
      - □ ContentResolver.update() ➔ Verb: update
      - □ ContentResolver.query() ➔ Verb: read
    - APIs:
      - □ External storage: FileOutputstream.write()➔ Verb: write
      - □ Other APIs: manually define verbs

Privacy Policy Generator



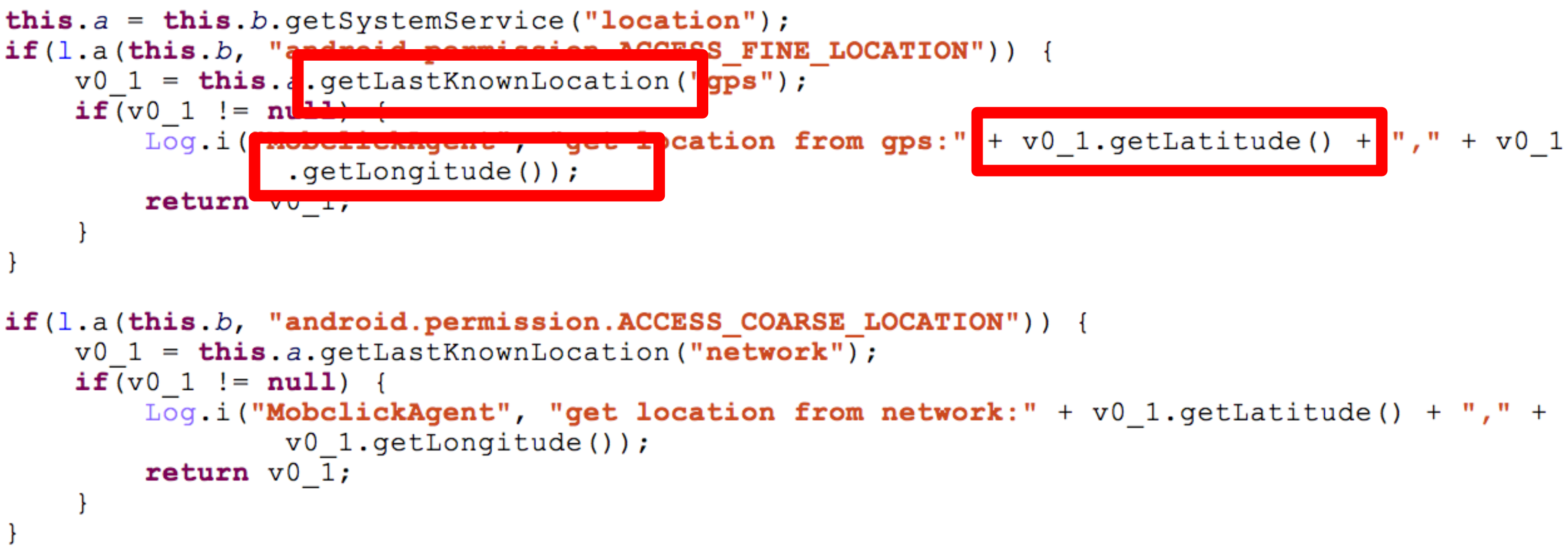
- Template of each generated sentence:
  - *Sentence* = [pre-condition] *subject verb object* [post-*condition*]
  - Object: Private information extracted from official description.
  - Pre-condition and post-condition:
    - Six kinds of condition identified in code.
      - □ UI events: add "when you press the button" as condition.
  - Information retained or not.
    - If the information is retained, we add additional sentence after it.
      - □ "This information will be retained in file/log"
      - □ "This information will be transferred out via SMS/internet/bluetooth"

Post-Process

- Remove duplicate sentences.
  - Different APIs get the same information.
    - getAccounts() and getAccountsByType().
  - The private information obtained by one API can be covered by another API.
    - getLatitude() and getLastKnownLocation().

Post-Process

- Example



```
this.a = this.b.getSystemService("location");
if(l.a(this.b, "android.permission.ACCESS_FINE_LOCATION")) {
    v0_1 = this.a.getLastKnownLocation("gps");
    if(v0_1 != null) {
        Log.i("MobclickAgent", "get location from gps:" + v0_1.getLatitude() + "," + v0_1
                .getLongitude());
        return v0_1;
    }
}

if(l.a(this.b, "android.permission.ACCESS_COARSE_LOCATION")) {
    v0_1 = this.a.getLastKnownLocation("network");
    if(v0_1 != null) {
        Log.i("MobclickAgent", "get location from network:" + v0_1.getLatitude() + "," +
                v0_1.getLongitude());
        return v0_1;
    }
}
```

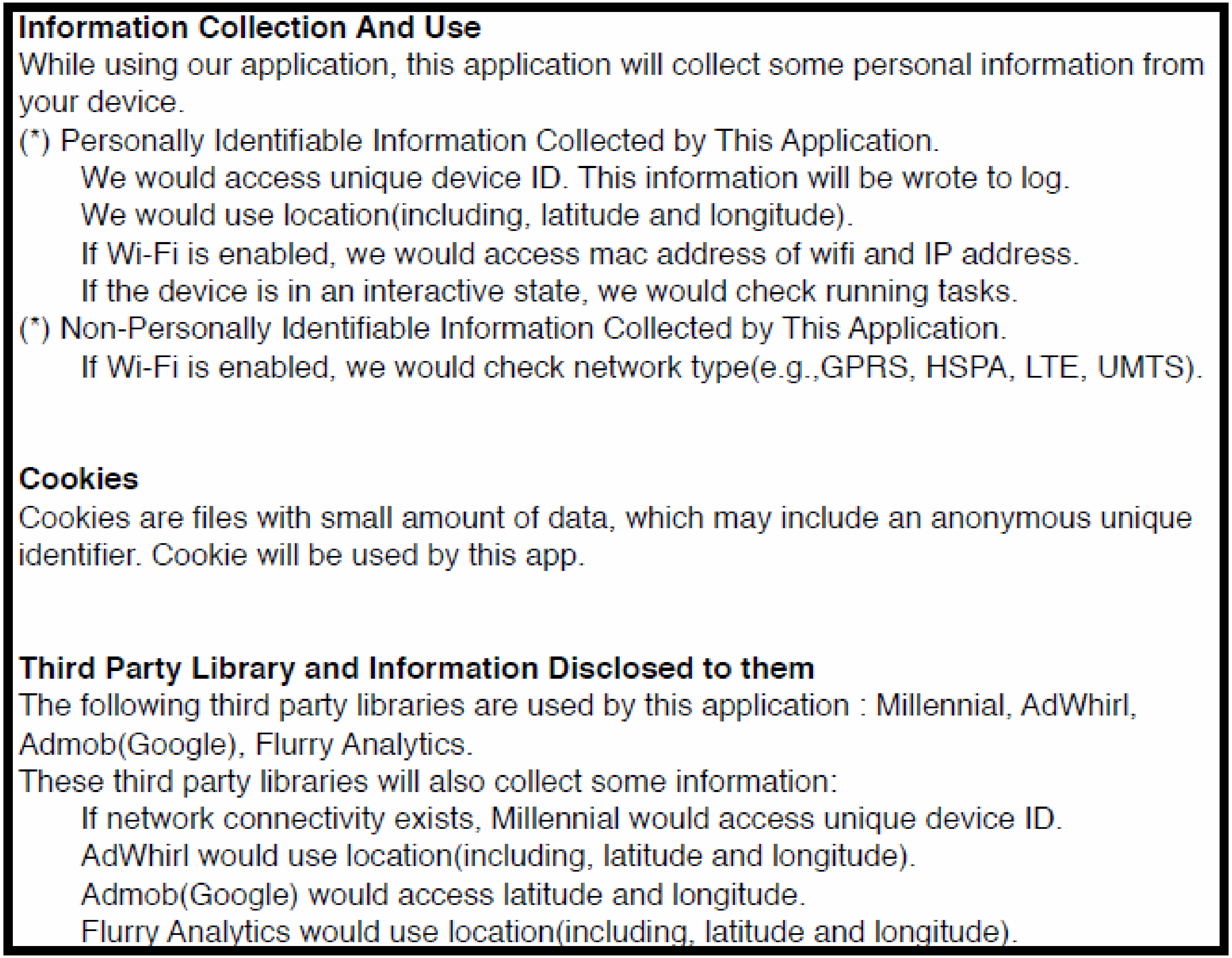
  - App calls getLastKnownLocation(), getLatitude(), and getLongitude() in the same method.
  - Only one sentence will be generated.
  - “We would use location (including, latitude and longitude).”

Post-Process

- Change the order of the remaining sentences.

  Sensitive behaviors (e.g., read contacts/SMS) are displayed first.

- Private information risk rank list:
  - 1 contact
  - 2 SMS
  - 3 call log
  - 4 browser history
  - 5 calendar
  - 6 device ID
  - 7 audio
  - 8 camera
  - 9 location
  - …



Information Collection And Use

While using our application, this application will collect some personal information from your device.

(*) Personally Identifiable Information Collected by This Application.

We would access unique device ID. This information will be wrote to log.
We would use location(including, latitude and longitude).
If Wi-Fi is enabled, we would access mac address of wifi and IP address.
If the device is in an interactive state, we would check running tasks.

(*) Non-Personally Identifiable Information Collected by This Application.

If Wi-Fi is enabled, we would check network type(e.g.,GPRS, HSPA, LTE, UMTS).

Cookies

Cookies are files with small amount of data, which may include an anonymous unique identifier. Cookie will be used by this app.

Third Party Library and Information Disclosed to them

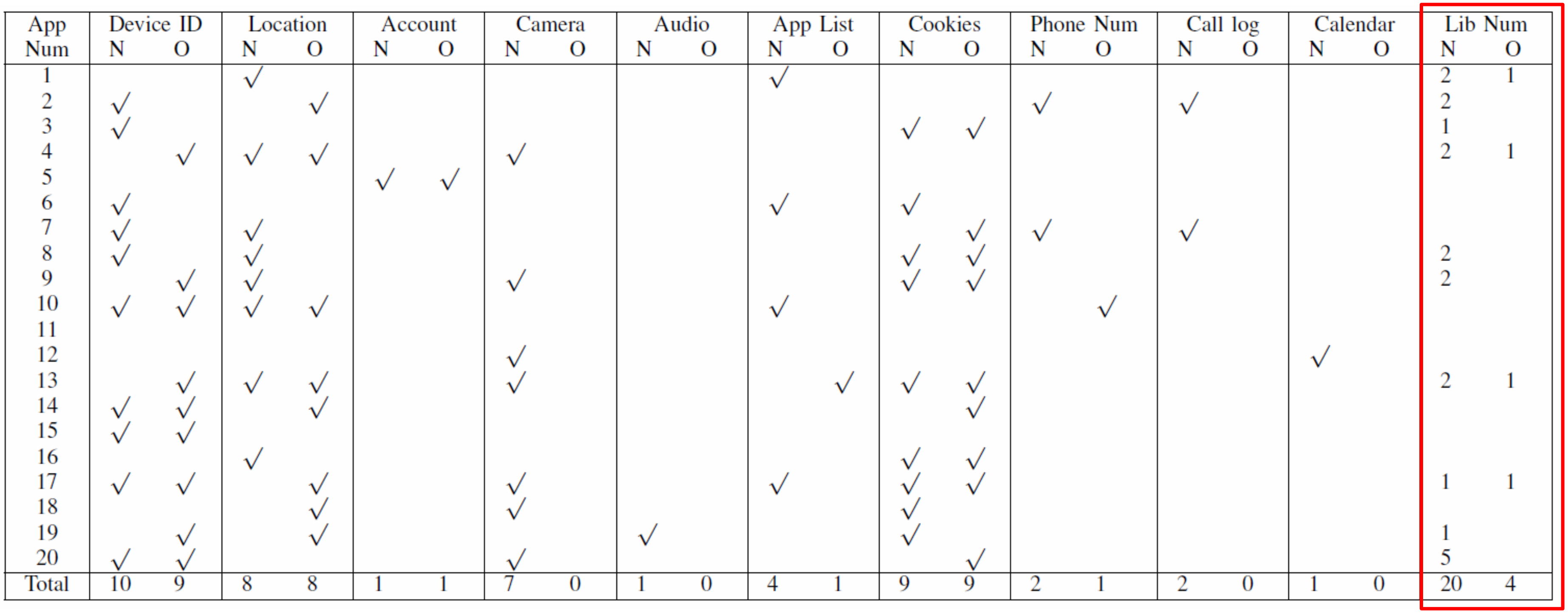
The following third party libraries are used by this application : Millennial, AdWhirl, Admob(Google), Flurry Analytics.

These third party libraries will also collect some information:

If network connectivity exists, Millennial would access unique device ID.
AdWhirl would use location(including, latitude and longitude).
Admob(Google) would access latitude and longitude.
Flurry Analytics would use location(including, latitude and longitude).

Evaluation

- Comparing the coverage of the privacy policies generated by our tool and that of existing privacy policies.
  - "N": privacy policies generated by our tool, "O": existing privacy policies
  - Existing privacy policies may be either incomplete or imprecise.



App	Device ID		Location		Account		Camera		Audio		App List		Cookies		Phone Num		Call log		Calendar		Lib Num	
Num	N	O	N	O	N	O	N	O	N	O	N	O	N	O	N	O	N	O	N	O	N	O
1			√								√										2	1
2	√			√											√		√				2	
3	√												√	√							1	
4		√	√	√			√														2	1
5					√	√																
6	√										√		√									
7	√		√											√	√		√					
8	√		√										√	√							2	
9		√	√				√						√	√							2	
10	√	√	√	√							√					√						
11																						
12							√												√			
13		√	√	√			√					√	√	√							2	1
14	√	√		√										√								
15	√	√																				
16			√										√	√								
17	√	√		√			√				√		√	√							1	1
18				√			√						√									
19		√		√					√				√								1	
20	√	√					√							√							5	
Total	10	9	8	8	1	1	7	0	1	0	4	1	9	9	2	1	2	0	1	0	20	4

Outline

- Spotting Issues in Apps’ Privacy Policies
- Generating Privacy Policies Templates for Apps
- Conclusion

Suggestions

- For normal users, please read the privacy policy before installing an app if it is available.
- For app developers, please provide clear privacy policies following the suggestions/guidelines, get familiar with the APIs/third-party libs used, and avoid over-claiming permissions.
- For companies that outsource the app development, please check the code and the privacy policy carefully before releasing the app.

Conclusion

- Correct and clear privacy policies are very useful to the apps.

- Identify three kinds of problems in privacy policies, and find many existing privacy policies have at least one problem.

- Develop PPChecker to automatically identify problems in an app's privacy policy by analyzing information from multiple sources.

- Develop AutoPPG to automatically generate privacy policy templates for apps *without* the need of source codes.

Thanks