# POSTER: The Popular Apps in Your Pocket Are Leaking Your Privacy

Xing Liu
Beijing Jiaotong University
Beijing, China, 100044
xingliu@bjtu.edu.cn

Wei Wang
Beijing Jiaotong University
Beijing, China, 100044
wangwei1@bjtu.edu.cn

Jiqiang Liu
Beijing Jiaotong University
Beijing, China, 100044
jqliu@bjtu.edu.cn

## ABSTRACT

Smartphone users are facing serious threat of privacy leakage. This privacy leakage is caused not only by malicious applications (apps), but also by the most popular apps in one's pocket. In this poster, we present our study on the issues of information leakage caused by the most widely used apps in Chinese app markets. Our goal is to find what information is exposed by each popular app, and then to focus on the following three questions in order to explore the influence of this kind of information leakage: (1) to what extent can the information leaked by an app be used to characterize the user's behaviors? (2) to what extent can the information leaked by a number of apps in the same smartphone be used to characterize the user's behaviors? and (3) whether the leaked information from a number of smartphones can be integrated to predict the social behaviors? Preliminary experimental results on the top 50 popular apps in Chinese app markets show the serious situation of this kind of information leakage.

## Categories and Subject Descriptors

K.4.1 [**COMPUTERS AND SOCIETY**]: Public Policy Issues—*Privacy*

## Keywords

Android; popular apps; privacy leakage

## 1. INTRODUCTION

In recent years, Android-powered smartphones have become very popular in both personal and business uses. According to a report from the International Data Corporation (IDC) [5], Android-powered smartphones dominate the market with a 78% share in the first quarter of 2015, while according to a statistical result from AppBrain [2], the number of apps in Google Play has reached 1.5 million. However, most of the Android apps are only installed by a few people. In contrast, a number of apps are very popular in all kinds of users, especially the apps in the top 50 list [1]. Most of users have installed more than one app in the top 50 list in their smartphones.

As a portable device, the smartphone stores a lot of personal information. The usage information of the smartphone also reflects the users' habits, interests or relationships. Hence, the privacy on the Android-powered smartphones is a big issue. In previous work, researchers made their efforts to detect malicious apps that steal users' privacy [6] and to develop tools [4][7] that are used to discover the privacy leakage paths in apps. However, there is little attention paid to privacy leakage caused by the most popular apps developed by reputed companies or groups. These apps are normally in the Antivirus companies' Whitelists and are not alerted. However, the developers of these popular apps also have the motivation to collect users' usage information of their apps in order to improve their apps' quality. If the collected information is not well protected, it can be exposed to any network sniffers between the apps and their servers, resulting in serious privacy leakage.

In this poster, we present our preliminary study on the information leakage issues caused by the most popular apps in Chinese app markets. We design a tool called *ILDDroid* (Information Leakage Discover Droid) to discover the private information leaked by the popular apps. ILDDroid tries to discover unknown private information leakage, not just the information protected by Android permissions. Based on the analysis results from ILDDroid, we aim to answer the following three questions to explore the influence of this kind of information leakage:

- to what extent can the user be identified or characterized with the information leaked by an app?
- to what extent can the user be identified or characterized with the information leaked by a number of apps in the same smartphone?
- is it possible to divide users into different groups and to predict the social behaviors of these groups based on the leaked information from a number of users (or smartphones)?

We conduct preliminary experiments on the top 50 popular apps in Chinese app markets. The experimental results show a serious situation of this kind of information leakage. The popular apps in one's pocket not only send out smartphones' hardware information that can be used to uniquely identify one's smartphones, but also expose when and where one uses these apps and even how long one spends on each

*CCS'15,* October 12–16, 2015, Denver, Colorado, USA.
ACM 978-1-4503-3832-5/15/10.
DOI: http://dx.doi.org/10.1145/2810103.2810127 .

item in one app. We believe that this information is enough to distinguish an individual user. More extensive experiments for groups' behaviors prediction are being conducted.

## 2. SYSTEM DESIGN

Different from the previous state-of-art Android analysis tools [3][4], ILDDroid tries to discover unknown private information leakage at the app's runtime. According to Zhou's [7] research results, it is possible to identify a user based on the leaked information that is not protected by Android permissions. Hence, ILDDroid captures the network traffic data between the apps and their servers at first. Then the captured data is analyzed to distinguish the data sent for asking services and the data sent for collecting users' usage information. We analyze the network traffic data sent for collecting users' usage information to discover what information they leak. For some apps that send encrypted data, we analyze the disassembled codes and running a modified version of these apps to find out what information is sent. ILDDroid can be divided into two parts: network traffic analyzer and app analyzer. We illustrate the system overview in Figure 1.

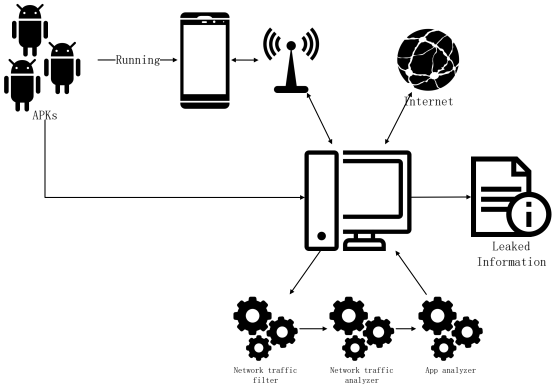


Figure 1: System Overview

### 2.1 Network Traffic Analyzer

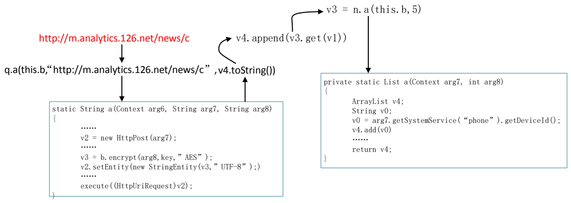
Network traffic analyzer captures the network traffic between the apps and their servers at first. We just capture the data transmitted through HTTP protocol. In our work, we set up a WiFi hotspot with a computer and run a network sniffer on that computer. An Android-powered smartphone is connected to this hotspot. Network traffic data generated by the apps running on the smartphone are totally logged in the computer. We run each app for several minutes and trigger most of their functions. After the network traffic data are logged, we find out the data transmitted for collecting users' usage information by the following rules: (1) Keywords in the target Urls, such as "register", "log" or "collect" *etc.*; (2) Keywords in the transmitted data, such as "time", "imei", or "action" *etc.*. This rule is not available for the encrypted data; (3) Returned data once data is sent. If the server returns nothing or just returns a status as "success" or "error" after the app send a lot of data, these data may contain users' usage information.

Once we find out the network traffic data sent for collecting users' usage information, we manually identify what privacy may be leaked.

### 2.2 App Analyzer

In order to discover the information transmitted by the encrypted data, we developed an app analyzer that performs both static analysis and dynamic analysis to find out what information is encrypted semi-automatically.

- Static analysis. The static analysis process is performed as shown in Figure 2. First, the analyzer finds out the Urls obtained from the network traffic analyzer in the apps' disassembled codes. Second, the analyzer identifies where these Urls are used. Third, the data sent with these Urls are tainted. Afterwards, static taint analysis is performed to discover the original data. This static analysis can find out many static data, such as the encrypted *IMEI, Network information etc.*. If there are some unknown method that invokes on the taint paths, we mark these points as interesting points and leave them to dynamic analysis.



```
http://m.analytics.126.net/news/c
q.a(this.b,"http://m.analytics.126.net/news/c" ,v4.toString())
v4. append(v3. get(v1))
v3 = n. a(this. b, 5)
```

```
static String a(Context arg6, String arg7, String arg8)
{
        ......
        v2 = new HttpPost(arg7);
        ......
        v3 = b. encrypt(arg8, key, " AES" );
        v2. setEntity(new StringEntity(v3, " UTF-8" );)
        ......
        execute((HttpUriRequest)v2);
}
```

```
private static List a(Context arg7, int arg8)
{
        ArrayList v4;
        String v0;
        v0 = arg7. getSystemService( "phone" ). getDeviceId();
        v4. add(v0)
        ......
        return v4;
}
```

Figure 2: Static analysis process

- Dynamic analysis. The dynamic analysis is performed based on the results of the static analysis. We inject some monitoring codes into the apps' disassembled codes at the points we are interested. The injected monitoring codes catch the runtime information we want and output it through the apps' log file. By running the repackaged modified apps, we understand what information is exposed.

## 3. EXPERIMENTAL RESULTS AND DISCUSSIONS

We conduct an experiment on the top 50 popular apps in Chinese app markets, each of them has more than 19 million monthly active users. We list some analysis results in Table 1.

The table shows the current serious situation of the information leakage caused by the popular apps. Most of the apps in the top 50 list transmit their collected users' information through unsafe ways, except the apps developed by big companies like *Tencent* and *Alibaba*. Some of these apps send the unencrypted collected data in JSON format directly, like "{"IMEI":351554053490182}". Some of them just simply change the label of the data, such as change the "{"IMEI":351554053490182}" to "{"td":351554053490182}". The other of them encrypt the collected data with AES. But the keys are generated from current time. It is easily compromised by analyzing the apps' disassembled codes.

What is worse, the information collected by these popular apps is far more than the users imagine. It not only contains the smartphone's hardware information, but also contains the detailed usage information as shown in Table

**Table 1: Information leaked in each app**

| App Name | Category | Target Urls | Leaked Information |
|---|---|---|---|
| Toutiao | News | oc.umeng.com/app_logs<br>log.snssdk.com/service/2/app_log | Address, Network status, Mac, Device id,<br>Device model, OS version, Display density,<br>Installed apps, App installed time, App launched time,<br>How long time the users spend on each activity of this app |
| Neteasy News | News | m.analytics.126.net/news/c | Address, Network status, Mac, IMEI, Device model,<br>OS version, CPU type, Display density, App launched time,<br>The time when the users open a category of news |
| WiFi Master Key | Tools | wifiapi02.51y5.net<br>woa.sdo.com/woa/datacollect/<br>mobads-logs.baidu.com | Address, Network status, IMEI, Device model, OS version,<br>Mac, Around wifi hotspots and their mac,<br>The time when the user is using this app |
| Xiaomi Market | Third party market | 123.129.202.147 | IMEI, Device model, OS version,<br>Installed apps and their version |
| UC Browser | Communication | track.uc.cn:9080/collect<br>utop.umengcloud.com | User id, Address, IP, Network status, Device model,<br>OS version Display density,<br>what the user browse and the corresponding time |
| TTPod | Music | collect.log.ttpod.com/ttpod_client_v2 | IMEI, Device model, OS version, CPU type, RAM size,<br>which song lists the users open and the corresponding time |

1. The usage information is generated by the apps and is not protected by Android permissions. It is therefore not cared by the third-party security companies. A single app may collect limited information from the users and may not leak users' privacy. However, multi-apps in the same smartphone actually leaks users' privacy. For example, if a user has installed the apps in Table 1, his address, reading interest and favorite songs are totally exposed to intentional monitors. The user's age can also be inferred.

Another question is the information leakage caused by the third-party analysis libraries in these apps. Apps package some third-party analysis libraries into themselves and send collected usage information to these third-party companies. These third-party companies analyze the collected data and give the developers a report of the usage of their apps. If a user has installed several apps containing the same third-party libraries, these third-party companies can easily obtain the usage information from different apps in one smartphone. The user's identity and other private information are thus leaked.

For the prediction of social behaviors, extended experiments are being conducted. We plan to extend our experiments to cover the top 200 popular apps in Chinese app markets. These top 200 apps almost cover every aspects of our daily life. In addition, we will capture the network traffic from a large number of users.

## 4. CONCLUSION

In this poster, we explore the information leakage caused by the most popular apps. We design a tool called *ILD-Droid* that is used to analyze what information is leaked by the widely used apps. The preliminary experimental results show that the user's privacy is leaked by the popular apps in pocket. We are conducting extensive experiments to study the feasibility of predicting the social behaviors of different groups of users.

## 5. ACKNOWLEDGEMENT

The work reported in this Poster was supported in part by the Fundamental Research Funds through the Central Universities of China under Grant No. K15JB00190, in part by Ph.D. Programs Foundation, Ministry of Education of China, under Grant 20120009120010, in part by the Scientific Research Foundation through the Returned Overseas Chinese Scholars, Ministry of Education of China, under Grant K14C300020, in part by the Program for Changjiang Scholars and Innovative Research Team in University under Grant IRT 201206, and in part by the 111 Project under Grant B14005.

## 6. REFERENCES

[1] Analysis. Top 50 apps in chinese app markets. `http://www.analysys.cn/`, 2015-05.

[2] AppBrain. Google play stats. `http://www.appbrain.com/stats/`, 2015-05-23.

[3] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. In *ACM SIGPLAN Notices*, volume 49, pages 259–269. ACM, 2014.

[4] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B. Chun, L. Cox, J. Jung, P. McDaniel, and A. Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)*, 32(2):5, 2014.

[5] I.D.Corporation. Smartphone os market share, q1 2015. `http://www.idc.com/prodserv/smartphone-os-market-share.jsp`, 2015-05.

[6] M. Zhang, Y. Duan, H. Yin, and Z. Zhao. Semantics-aware android malware classification using weighted contextual api dependency graphs. In *CCS2014*, pages 1105–1116. ACM, 2014.

[7] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. Gunter, and K. Nahrstedt. Identity, location, disease and more: Inferring your secrets from android public resources. In *CCS2013*, pages 1017–1028. ACM, 2013.