Quokka

TECHNICAL PAPER

Resurrecting the READ_LOGS Permission

on Samsung Devices

Ryan Johnson, Kryptowire / George Mason University
Angelos Stavrou, Kryptowire

Resurrecting the READ_LOGS Permission on Samsung Devices

Abstract

We have discovered an attack that allows a non-privileged application to continually force the generation and logging of sensitive process information in a readable log file using the /system/bin/dumpstate binary on Samsung Android devices. The log output of the dumpstate binary includes the Android log, kernel log, and other process-dependent log data. However, starting with Android 4.1, reading the Android log is no longer permitted to user applications because the READ_LOGS permission was considered to be a dangerous permission, but we were able to circumvent this limitation. To achieve this, we crafted an exploit that requires an application with the seemingly innocuous android.permission.RECEIVE_BOOT_COMPLETED permission. Reading the Android log empowers a non-privileged user application to obtain private data circumventing all permission checks. The approach to obtain the Android log data has worked on all Samsung devices we have examined ranging from the Samsung Galaxy S II up to and including the Samsung Galaxy S5 and the Samsung Note 4. The Android log generally contains private data written by the Android Operating System (OS), Google applications, and user applications. Moreover, we have identified 26 Samsung builds for Android where the Android OS writes the text of notifications by default to the Android log. Using our attacks on these 26 builds, we are able to get access to Facebook Messenger messages, text messages (including password resets), Google Chat messages, WhatsApp messages, missed calls, turn-by-turn directions from Google Maps, the sender and subject of emails, and any other notification. Our proof-of-concept application can obtain the text from all notifications that the Android OS receives for these builds. This enables a user application to obtain immensely private data from the user of these vulnerable Samsung devices. The vulnerable builds are for the previous generation of Samsung devices that are still currently being sold in retail stores (e.g., Samsung Galaxy S4, Samsung Note 3, Samsung Note Pro 12.2, etc.).

Vulnerability 1 – Reading the Android Log

Background

This vulnerability enables a user application to obtain the same functionality as the android.permission.READ_LOGS permission. The android.permission.READ_LOGS permission is not granted to user applications starting with Android 4.1, although it is granted to system applications and Android Debugging Bridge (ADB) [1]. The developers of the Android Operating System (OS) gave the android.permission.READ_LOGS permission a protection level of system|signature|development since user applications, Android OS processes, and Google applications can write sensitive data to the Android log [2]. There is a few minutes delay from when messages are written to the Android log and when they can be obtained by the user application with our approach. This vulnerability has been present on every Samsung Android device we have tested. This ranged from a Samsung Galaxy S II running Android 2.3 to a Samsung Galaxy S5 running Android 4.4.4.

This vulnerability can be used to gain certain sensitive data that would otherwise require multiple permissions to be declared in a user application's AndroidManifest.xml file. Some examples of sensitive data written to the Android log by Android OS processes, Google applications, and

Samsung applications are the user's email addresses, cell-tower ID (which can get an approximation of their location), raw GPS data, non-Google account names (e.g., Twitter), cellular network and Wi-Fi information, voicemail number, the name associated with a Gmail account, Google IDs (which can be used to locate a user's Google+ page), Integrated Circuit Card Identifier (ICCID), International Mobile Subscriber Identity (IMSI), Mobile Country Code Mobile Network Code (MCCMNC), MAC address, device serial number, URLs and GPS coordinates opened via an Intent with the android.intent.action.VIEW action string (which is common in the Facebook application), URLs in Chrome for which there are errors, which applications the user or system executes and when, and being able to tell whether the user is present or not. Appendix A contains numerous concrete anonymized examples of these items as they appear in the Android log.

User applications can have the insecure programming practice of writing sensitive information to the Android log [3]. We did not focus much on user applications, although we present a fairly egregious example of leaking data to the Android log from the iTriage application (com.healthagen.iTriage) in Appendix B. This user application wrote the username, password, cookies, insurance information, the user's medical procedures, the user's medical conditions, and the user's medications to the Android log.

Obtaining the Android Log Data

All the Samsung devices that we have examined write Android log data to the /data/log directory via the /system/bin/dumpstate binary when there is one of the following events: uncaught exception in an application's Dalvik bytecode, the application becomes unresponsive for a period of time, or when an error is encountered during the execution of an application's native code library. Any of these events will force the creation of a dumpstate file (e.g., /data/log/dumpstate_app_native.txt.gz) that contains the Android log, kernel log, system properties, network routing information, data from the proc file system, and additional low-level system data [4]. A dumpstate file size generally ranges between 2 to 7 MBs. A dumpstate file will contain the system, radio, and events buffers from the Android log. Once a resulting dumpstate file is created, a user application can decompress the file and examine its contents as a plaintext file. When one of the three aforementioned conditions occurs, this triggers the execution of the dumpstate binary by the /system/bin/debuggerd process with the flags and arguments shown below.

```
10-25 16:09:41.289   267   267 I DEBUG   : !@dumpstate -k -t -z -d -o
/data/log/dumpstate_app_native -m 8028
```

All of the examples in this document were produced on a Samsung Galaxy S4 running Android 4.4.2 with a build number of KOT49H.I337UCUFNI1 unless specifically noted otherwise. KOT49H.I337UCUFNI1 is a stock Samsung Android build with AT&T as the carrier. The dumpstate binary is executed by the /system/bin/debuggerd process which is owned by the root user. A normal user process will not be able to get the appropriate output when attempting to execute the dumpstate binary; it needs to be executed by a process with higher privileges such as the shell or root user. The debuggerd process establishes the default signal handlers for processes running on the device. When a signal is received, the debuggerd process will attach to the process using ptrace and obtain information from the process. In Samsung Android builds, the dumpstate binary is also executed. The process information below and the logcat entry above are correlated using the process ID (PID). The PID of the /system/bin/debuggerd process is 267 in this context.

```
root      267   1     1180   596        -16     1            ffffffff 00000000
S /system/bin/debuggerd
```

The /data/log directory gets created in Samsung's init.rc file. During the boot process, the instructions in the init.rc file will be executed. Below is a snippet of the init.rc file from a Samsung Galaxy S4 running Android 4.4.2 with a build number of KOT49H.I9500XXUGNJ1.

```
# SA, System SW, SAMSUNG create log directory
    mkdir /data/log 0775 system log
    chown system log /data/log
    mkdir /data/anr 0775 system system
    chown system system /data/anr
    chmod 0775 /data/log
    chmod 0775 /data/anr
    restorecon /data/log
    restorecon /data/anr
```

Below is a listing of all files and their respective file permissions in the /data/log directory. The file permissions show that any user on the device has read access to various files including the dumpstate files colored in red. A file is world-readable if an r appears in the third column from the right (e.g., -rw-r--r--) for the file permission listing.

```
-rw-r--r-- u0_a239  u0_a239       697 2014-08-29 20:11 CallDropInfoLog.txt
-rw------- system   system        233 2014-09-24 16:24 ContainerHistory.txt
-rw------- system   system      22498 2014-09-24 16:23 PreloadInstaller.txt
-rw----r-- system   system        512 2014-10-23 11:36 Status.dat
-rw-r--r-- shell    log       1007511 2014-10-12 15:51 dumpstate_app_anr.txt.gz
-rw-r--r-- shell    log        605572 2014-10-18 20:54 dumpstate_app_error.txt.gz
-rw-r--r-- shell    log        568151 2014-10-20 23:37 dumpstate_app_native.txt.gz
-rwx------ system   system         25 2014-10-23 09:30 gyroOffset
-r--r--r-- root     root            0 2013-09-30 19:08 lock
-rw------- system   system       2431 2014-10-12 15:51 looper.txt
-rw------- system   system      47376 2014-10-23 11:35 omc.log
-rw-rw---- system   system       6984 2014-10-23 11:02 power_off_reset_reason.txt
-rw-r--r-- system   system       3403 2014-03-04 16:36 poweroff_info.txt
-rw-r--r-- system   system       1307 2014-05-20 13:28 powerreset_info.txt
-rw-r--r-- system   system     131118 1971-01-06 17:53 recovery_kernel_log.txt
-rw-r--r-- system   system     932054 1971-01-06 17:53 recovery_last_kernel_log.txt
-rw-r--r-- system   system     224033 1971-01-06 17:53 recovery_log.txt
-rw-r--r-- system   system       1000 2014-09-24 16:20 recovery_patch_log.txt
-rw------- system   system    5704942 2014-10-23 14:34 setupwizard.txt
```

It is easy for a user application to cause any of the three required conditions to make the /system/bin/debuggerd process execute the dumpstate binary. To generate the /data/log/dumpstate_app_error.txt.gz file, a user application can crash itself by throwing an uncaught runtime exception (e.g., java.lang.NullPointerException). This situation will create a system message indicating the name of the application that has crashed which may alert the user. To generate the /data/log/dumpstate_app_anr.txt.gz file, a user application needs to create an Application Not Responding (ANR) event. The application can sleep for a period of time on its main (i.e., user interface) thread to create an ANR event, although the amount of time it takes seems to be variable and considerably longer than the 5 seconds as stated on the Android Developers website [5]. The ANR event will generate a system message identifying the name of the application that is not responding. To generate the /data/log/dumpstate_app_native.txt.gz file, a user app can encounter an error during the execution of one of its native code libraries. This will not alert the user that an error has occurred in native code if done in a particular way so that it does not propagate back to the Dalvik Virtual Machine (VM). Appendix C contains screenshots of the result of the three events that trigger the creation of a dumpstate file.

We have identified an appropriate error that will not crash the entire user application or create any visual alert for the user to notice. We use Java Native Interface (JNI) to call a C function in an app's native code library. In the C function, the process is forked. The child process calls the

abort function and the parent process simply returns. If the process is not forked before calling the abort function, the entire application will crash. The abort function will send the SIGABRT signal [6]. The SIGABRT will be received by the /system/bin/debuggerd process which will execute the /system/bin/dumpstate binary. Encountering an error in an app's native code library is the preferred approach since the user is not alerted with a system message, it does not leave a crashed or unresponsive application in the recent applications list, and it can be done stealthily in the background. The approach is performed by a service (i.e., android.app.Service) application component, so that the user does not need to actually be using the application to create the circumstance to generate the dumpstate_app_native.txt.gz file. Therefore, the service can always be running in the background due to the android.permission.RECEIVE_BOOT_COMPLETED permission and periodically call the C function to trigger the creation of the dumpstate_app_native.txt.gz file at some regular interval. The user app can then exfiltrate the dumpstate_app_native.txt.gz file itself or process and filter it locally prior to exfiltration. Exfiltrating the data without the android.permission.INTERNET permission requires that the data be sent using an android.content.Intent object with the android.intent.action.VIEW action string to the browser and have the data to exfiltrated be encoded in a query string of a URL [7]. This will open the browser and may raise the suspicion of the user. It would be easier to request the android.permission.INTERNET permission and just send it to a remote server.

Threat Model

The attack assumes that the user has downloaded an application that requires the android.permission.RECEIVE_BOOT_COMPLETED permission and possibly the android.permission.INTERNET for easier exfiltration. The application will be able to persistently run via a service application component. The application will use JNI to call a native library written in C which will cause an error outside of the Dalvik VM to generate new instances of the /data/log/dumpstate_app_native.txt.gz file. The JNI call to the native library will be executed periodically in the background from the application's service application component. It can then decompress the /data/log/dumpstate_app_native.txt.gz file to read the Android logs. The application can then optionally process the log and use regular expressions to filter the data and send the data over a network socket.

Threat Resolution

If the various dumpstate files located in the /data/log directory were not world-readable, then the approach to obtain the Android log data would not be viable since the file permissions would not allow these files to be accessed by user applications. So simply changing these files to not be world-readable would be sufficient to prevent the attack. Crash dumps are important, but it is dangerous to make them available to every user on the device. The dumpstate files have shell as the file owner and log as the group owner, so the device owner could still pull them off the device using ADB if they were changed to not be world-readable.

```
-rw-r--r-- shell    log       1007511 2014-10-12 15:51 dumpstate_app_anr.txt.gz
-rw-r--r-- shell    log        605572 2014-10-18 20:54 dumpstate_app_error.txt.gz
-rw-r--r-- shell    log        568151 2014-10-20 23:37 dumpstate_app_native.txt.gz
```

To successfully prevent user apps from accessing these files on a non-rooted device, the user would need to use ADB shell only once to enter some commands to prevent them from being world-readable permanently. We will focus on the /data/log/dumpstate_app_native.txt.gz file, but the same approach works for the other dumpstate files in the /data/log directory. The user would need to use ADB shell to cd to the /data/log directory. Then they would need to delete the /data/log/dumpstate_app_native.txt.gz file if it exists. They would then need to use the touch

command to create the dumpstate_app_native.txt.gz and dumpstate_app_native.txt.gz.tmp files. Then change the file permissions on both files by using the chmod command to set the files to be readable, writeable, and executable by no one (i.e., chmod 000 <file name>). When an application experiences a crash in its native code library, the dumpstate binary will not overwrite or write to either of these files due to the restrictive file permissions, thus preventing the output from dumpstate binary to be obtained by user applications. The dumpstate binary initially executes as the root user, but it drops root privileges. The process changes to the shell user and the log group as well as a few other groups. The dumpstate process executes as the shell user and uses output redirection when writing to the file instead of deleting it first, so the dumpstate binary will be denied access to the file due the extremely restrictive file permissions.

Vulnerability 2 – Reading Android Notifications from the Android Log

Background

The second vulnerability is that the text of Android notifications is written to the Android log as they are received in certain Samsung Android builds. This behavior appears not be present prior to Android 4.3 except for Android 4.1.2 and has been fixed in Android 4.4.4. All of the vulnerable builds we have tested and verified are from Android 4.1.2, Android 4.3, and Android 4.4.2. All notifications on these vulnerable builds are written to the Android log with some notable ones being Facebook Messenger messages, text messages (including password resets), Google Chat messages, WhatsApp messages, missed calls, turn-by-turn directions from Google Maps, and the sender and subject of emails. We rely on the first vulnerability to be able to read the Android log to get the text information from notifications.

Samsung introduced this vulnerability by adding some functionality to the standard Android Open Source Project (AOSP) code for the android.app.Notification class in the Android framework. In the vulnerable Samsung builds, the android.app.Notification class has a much more verbose version of the android.app.Notification.toString method than the corresponding AOSP version of the android.app.Notification class [8]. The more verbose Samsung version includes the following instance variables of the android.app.Notificaiton object in its toString method: contentTitle, contentText, and tickerText. These instance variables contain the text of the notification. Appendix D contains a snippet of the android.app.Notification.toString method in smali format where these instance variables are being added to a java.lang.StringBuilder object that is being used to generate the string representation of the Notification object. We pulled the /system/framework/framework.odex file from the device and used baksmali [9] to convert the odex file into a directory of hierarchical smali files. Below is the toString output of an android.app.Notification object from a vulnerable Samsung build (KOT49H.I337UCUFNI1) running Android 4.4.2.

```
Notification(pri=0 icon=7f020000 contentView=com.kryptowire.bha/0x1090086
vibrate=null sound=null defaults=0x0 flags=0x0 when=1415746428600
ledARGB=0x0 contentIntent=N deleteIntent=N contentTitle=Generic
Title contentText=Generic Subject tickerText=Here is a Message kind=[null])
```

Below is the output of the android.app.Notification.toString method With the same exact same notification from the AOSP version of the android.app.Notification class from a non-Samsung device running Android 4.4.2.

```
Notification(pri=0 contentView=com.kryptowire.bha/0x1090064 vibrate=null
sound=null defaults=0x0 flags=0x0 kind=[null])
```

In the com.android.server.NotificationManagerService.enqueueNotificationInternal method, the android.app.Notification.toString method gets called whenever a notification is received from an application or the Android OS itself. Then the string representation of the Notification, as well as a few other data items, are passed as parameters to the android.util.EventLog.writeEvent(int tag, Object... list) method. The parameters to the android.util.EventLog.writeEvent(int tag, Object... list) method call end up in the events buffer of the Android log with a log tag of notification_enqueue. Below is a snippet from the Android 4.4.4 AOSP source code for the com.android.server.NotificationManagerService.enqueueNotificationInternal method [10].

```
// This conditional is a dirty hack to limit the logging done on
// behalf of the download manager without affecting other apps.
if (!pkg.equals("com.android.providers.downloads") || Log.isLoggable("DownloadManager",
Log.VERBOSE)) {
    EventLog.writeEvent(EventLogTags.NOTIFICATION_ENQUEUE, pkg, id, tag, userId,
notification.toString());
}
```

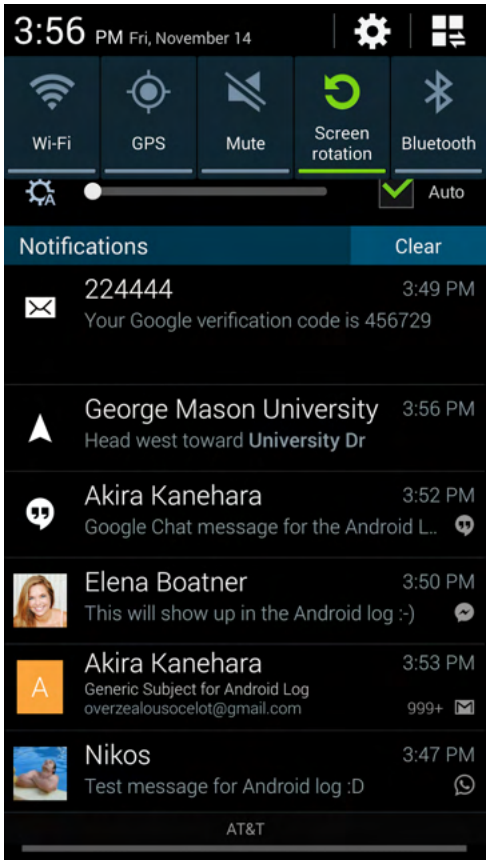
The first parameter to android.util.EventLog.writeEvent(int tag, Object... list) method determines the log tag. We pulled the /system/framework/services.odex file from the device to examine what integer corresponded to the first parameter (i.e., EventLogTags.NOTIFICATION_ENQUEUE) to the android.util.EventLog.writeEvent(int tag, Object... list) method. We used baksmali to disassemble the services.odex file into smali files. We then examined the com/android/server/EventLogTags.smali file to get the value of EventLogTags.NOTIFICATION_ENQUEUE integer constant, which is shown below in smali format.

```
.field public static final NOTIFICATION_ENQUEUE:I = 0xabe
```

This was also further verified by examining the /system/etc/event-log-tags file which contains the mapping of integer to string values used in conjunction with the android.util.EventLog class [11]. We confirmed this by writing some events using the android.util.EventLog.writeEvent(int tag, Object... list) method with a tag value of a decimal value of 2750 (i.e., 0xabe in hex) and the resulting log tag of notification_enqueue appeared as the log tag in the Android log. In addition, anything in the second parameter (i.e., Object… list) also appeared in the Android log on our Samsung Android device running 4.4.2 (KOT49H.I337UCUFNI1).

Android notifications

Below is a screenshot from a Samsung Galaxy S4 device running Android 4.4.2 with a build number of KOT49H.I337UCUFNI1 showing notifications from various applications. In order from top to bottom are notifications from SMS (com.android.mms), Maps (com.google.android.apps.maps), Hangouts (com.google.android.talk), Facebook Messenger (com.facebook.orca), Gmail (com.google.android.gm), and WhatsApp (com.whatsapp). Next to the screenshot is the accompanying message for the notification as it shows up in the Android log from the /data/log/dumpstate_app_native.txt.gz file. In addition, any notification will have its text written to the Android log on the vulnerable Samsung builds. We are just highlighting the commonly used apps that contain private data. Appendix E contains notifications from various apps as they appear in the Android log. Some of the data from notifications in Appendix E can be obtained by requesting the appropriate permissions (i.e., data for missed calls, text messages, and network information), but some data cannot (e.g., Facebook messages, Google Chat messages, WhatsApp messages, and the sender’s name and subject from email clients).




```
11-14 15:49:07.983   813  1163 I
notification_enqueue:
[com.android.mms,123,NULL,0,Notification(pri=2
icon=7f0202c3
contentView=com.android.mms/0x1090086 vibrate=null
sound=content://settings/system/notification_sound
defaults=0x4 flags=0x1 when=1415998146000
ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=224444 contentText=Your Google
verification code is 456729 tickerText=224444:
,ÄéYour Google verification code is 456729
kind=[android.message])]

11-14 15:56:44.792   813  1438 I
notification_enqueue:
[com.google.android.apps.maps,39796916,NULL,0,Noti
fication(pri=1 icon=7f02032a
contentView=com.google.android.apps.maps/0x1090086
vibrate=null sound=null defaults=0x0 flags=0x2
when=1415998484779 ledARGB=0x0 contentIntent=Y
deleteIntent=N contentTitle=George Mason
University contentText=Head west toward University
Dr tickerText=N kind=[null] 1 action)]

11-14 15:52:11.512   813  1438 I
notification_enqueue:
[com.google.android.talk,0,com.google.android.talk
:chat:1,0,Notification(pri=1 icon=7f020573
contentView=com.google.android.talk/0x1090086
vibrate=default
sound=android.resource://com.google.android.talk/r
aw/2131230729 defaults=0x6 flags=0x11
when=1415998331013 ledARGB=0x0 contentIntent=Y
deleteIntent=Y contentTitle=Akira Kanehara
contentText=Google Chat message for the Android
Log :-) tickerText=Akira Kanehara: Google Chat
message for the Android Log :-) kind=[null])]

11-14 15:50:02.517   813  1391 I notification_enqueue:
[com.facebook.orca,10000,t_mid.1415995249174:9405c32ed74e69fa79,0,Notification(pri=1
icon=7f0207a9 contentView=com.facebook.orca/0x1090086 vibrate=null sound=null
defaults=0x0 flags=0x1 when=1415998201262 ledARGB=0xff00ff00 contentIntent=Y
deleteIntent=N contentTitle=Elena Boatner contentText=This will show up in the Android
log :-) tickerText=Elena Boatner: This will show up in the Android log :-) kind=[null] 1
action)]

11-14 15:53:44.143   813 22582 I notification_enqueue: [com.google.android.gm,-
1847466507,NULL,0,Notification(pri=0 icon=7f0200f6
contentView=com.google.android.gm/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x11
when=1415998423942 ledARGB=0x0 contentIntent=Y deleteIntent=Y contentTitle=Akira Kanehara
contentText=Generic Subject for Android Log tickerText=Akira Kanehara kind=[null] 2
actions)]

11-14 15:47:29.547   813   824 I notification_enqueue:
[com.whatsapp,1,NULL,0,Notification(pri=0 icon=7f0205c2
contentView=com.whatsapp/0x1090086 vibrate=null sound=null defaults=0x0 flags=0x0
when=1415998049288 ledARGB=0x0 contentIntent=Y deleteIntent=N contentTitle=Nikos
contentText=Test message for Android log :D tickerText=Message from Nikos kind=[null])]
```

List of Vulnerable Builds

We tested the devices we were able to physically obtain, and we also downloaded stock images for various Samsung Android builds and tested them to see if the text of notifications appeared in the Android log. All of the vulnerable builds that we encountered were from Android 4.1.2, Android 4.3, and Android 4.4.2. It seems likely that many of the Android 4.3 and Android 4.4.2

builds except for the ones that have come out recently may also be vulnerable, although we have not verified this through testing. Below is a list of vulnerable builds that we have found.

Device	OS Version	Build Number
Samsung GS4	4.4.2	KOT49H.I337UCUFNI1
Samsung GS4	4.4.2	KOT49H.I545VRUFNC5
Samsung GS4	4.4.2	KOT49H.I337UCUFNB1
Samsung GS4	4.4.2	KOT49H.L720VPUFNAE
Samsung GS4	4.4.2	KOT49H.I9500XXUFNE7
Samsung GS4	4.4.2	KOT49H.I9500ZSUDNF2
Samsung GS4	4.4.2	KOT49H.I9500ZSUDNF1
Samsung GS4	4.4.2	KOT49H.I9500ZSUDNB3
Samsung GS4	4.4.2	KOT49H.I9500XXUFNB7
Samsung GS4	4.4.2	KOT49H.I9500XXUGNI1
Samsung GS4	4.4.2	KOT49H.I9500XXUGNJ1
Samsung Tab Pro 12.2	4.4.2	KOT49H.T900UEUANB5
Samsung Tab Pro 12.2	4.4.2	KOT49H.P907AUCU1AND7
Samsung Tab Pro 12.2	4.4.2	KOT49H.T900UEUAND4
Samsung Note 3	4.4.2	KOT49H.N900AUCUCNC2
Samsung Note 3	4.3	JSS15J.N900XXXUBMHC_LLK
Samsung Note 3	4.3	JSS15J.N900AUCUBNB4
Samsung GS4	4.3	JSS15J.L720VPUEMK2
Samsung GS4	4.3	JSS15J.I9500XXUEMK8
Samsung GS4	4.3	JSS15J.I9500ZSUCMK3
Samsung GS4	4.3	JSS15J.I9500ZSUCMJ6
Samsung GS4	4.3	JSS15J.I9500UBUEMK1
Samsung GS4	4.3	JSS15J.I9500XXUEMJ5
Samsung GS4	4.3	JSS15J.I9500XXUEMJ8
Samsung GS3	4.3	JSS15J.L710VPUCMK3
Samsung GS3	4.3	JSS15J.1535VRUCNC1
Samsung Galaxy Pocket Neo	4.1.2	JZO54K.S5310XXAME2

Threat Resolution

This vulnerability is not present in Samsung Android 4.4.4 and Android 5 builds. We looked at the Android framework for a Samsung Android 4.4.4 device. We noticed that the android.app.Notification class has a static boolean variable named DEBUG which controls whether or not the actual strings for the contentTitle, contentText, and tickerText instance variables get included in the output of its toString method. For the existing vulnerable builds from Android 4.1.2, Android 4.3, and Android 4.4.2, the resolution to the first vulnerability also fixes this vulnerability. The second vulnerability is dependent on the first vulnerability since Android log access is required to obtain the text of Android notifications that are written to the Android

log. So making the dumpstate files (e.g., dumpstate_app_native.txt.gz) not world-readable would alleviate both vulnerabilities. If possible, a device running a vulnerable build should be updated to a Samsung build running Android 4.4.4 or higher.

Appendix A: Anonymized Data As It Appears in the Android Log from a Samsung Build

<Data obtained> <process name that wrote the log message>
<Android log message(s) of the format (timestamp, PID, PPID, Log Level, Log Tag, Log Message)>

Android OS processes and Google Applications:

Voicemail Number (com.android.phone):

```
10-24 12:09:04.809  1362 13208 D SIMRecords: getVoiceMailNumber()  voiceMailNum
+18042487645SIMID0

10-24 15:56:57.591  1359  1359 D SIMRecords: VM: ADN Record 'Voicemail'
'+18042487645 null' EF[MBDN]
```

Gmail account (system_server):

```
10-13 16:51:53.898   812  1371 I am_create_activity:
[0,1117996024,253,com.google.android.gm/.ConversationListActivityGmail,android.
intent.action.VIEW,application/gmail-
ls,content://com.android.gmail.ui/whymewhynot@gmail.com/conversation/1481882726
767452911?appVersion=4900120&folderUri=content://com.android.gmail.ui/whymewhyn
ot @gmail.com/label/%5Esq_ig_i_personal,268484608,com.google.android.gm]
```

Gmail account and Google ID (can be used to view a Google Plus page https://plus.google.com/105708451973236075042) (com.google.process.gapps)

```
10-25 14:46:12.983  1598  3207 I GLSUser : [p] Fetched account id for
whymewhynot@gmail.com : 105708451973236075042

10-25 14:46:14.704  1598 13230 I GLSUser : [p] Fetched account id for
feelgoodman@gmail.com : 101542254781151609167

10-25 14:46:14.674  1598  4312 I GLSUser : [p] Fetched account id for
toomuchisnotenough@gmail.com : 112085291016639688831
```

Enumeration of Gmail accounts on the device (com.google.process.gapps)

```
10-25 14:45:51.292  1598 18075 V GLSUser : [ AccountStateSummary ] - get() -
<StoredState>[isConsistent= true, expectedNames= whymewhynot@gmail.com,
toomuchisnotenough @gmail.com,feelsgoodman@gmail.com, expectedHmacs=*****]
```

Gmail account (com.google.android.syncadapters.calendar)

```
10-25 15:30:52.257 27324 31291 V CalendarSyncAdapter: Saving inProgress state:
{calendarId= whymewhynot@gmail.com, maxAttendees=50, maxResults=200,
timeMax=2015-11-01T00:00:00.000Z, updatedMin=2014-10-25T19:10:13.054Z}

10-25 15:30:52.537 27324 31297 V CalendarSyncAdapter: Saving inProgress state:
{calendarId=toomuchisnotenough@gmail.com, maxAttendees=50, maxResults=200,
timeMax=2015-11-01T00:00:00.000Z, updatedMin=2014-10-24T00:00:21.882Z}

10-25 17:03:11.971 18523 18554 D CalendarSyncAdapter: Changed feedId ->
whymewhynot@gmail.com
```

Google ID (105708451973236075042) (system_server)

```
10-25 14:32:40.830   808   808 I notification_cancel:
[com.google.android.apps.plus,2131558555,com.google.android.apps.plus:notificat
ions: 105708451973236075042,0,0,64]

10-25 14:32:40.380   808   808 I notification_cancel:
[com.google.android.apps.plus,2131558559,com.google.android.apps.plus:notificat
ions: 101542254781151609167,0,0,64]
```

WiFi Information (system_server):

```
10-20 23:23:09.446   810  1090 D ConnectivityService: Captive portal check
NetworkInfo: type: WIFI[], state: CONNECTING/CAPTIVE_PORTAL_CHECK, reason:
(unspecified), extra: "NETGEAR242", roaming: false, failover: false,
isAvailable: true, isConnectedToProvisioningNetwork: false

10-20 23:23:09.446   810  1090 D ConnectivityService:
handleCaptivePortalTrackerCheck: call captivePortalCheckComplete
ni=NetworkInfo: type: WIFI[], state: CONNECTING/CAPTIVE_PORTAL_CHECK, reason:
(unspecified), extra: "NETGEAR242", roaming: false, failover: false,
isAvailable: true, isConnectedToProvisioningNetwork: false

10-24 17:34:21.995   812  1051 D ConnectivityService: Captive portal check
NetworkInfo: type: WIFI[], state: CONNECTING/CAPTIVE_PORTAL_CHECK, reason:
(unspecified), extra: "korax-5GHz", roaming: false, failover: false,
isAvailable: true, isConnectedToProvisioningNetwork: false

10-25 16:09:32.050   808   808 D LocSvc_java: updateNetworkState available
info: NetworkInfo: type: WIFI[], state: CONNECTED/CONNECTED, reason:
(unspecified), extra: "NETGEAR242", roaming: false, failover: false,
isAvailable: true, isConnectedToProvisioningNetwork: false
```

Non-Google account (Twitter user name can be used to view profile https://twitter.com/atrandom) (system_server)

```
10-09 08:01:12.797   849  1074 I power_partial_wake_state:
[1,*sync*/com.twitter.android.provider.TwitterProvider/com.twitter.android.auth
.login/ atrandom]
```

Private IPv4 address on the carrier network (com.android.phone)

```
10-25 18:45:53.663  1361  1528 D DataCallResponse: addr/pl=10.155.221.108/32

10-25 14:00:40.355  1361  1528 D DC-1    : REQ_GET_LINK_PROPERTIES
linkProperties{InterfaceName: rmnet_usb0 LinkAddresses: [10.155.221.108/32,]
Routes: [0.0.0.0/0 -> 10.155.221.109,] DnsAddresses: [172.26.38.1,172.26.38.2,]
Domains: nullMTU: 0}
```

Integrated Circuit Card Identifier (ICCID) (com.android.phone)

```
10-25 14:00:37.693  1361  1361 D SIMRecords: mIccId: 89014503281383474393
10-25 14:00:37.693  1361  1361 D SIMRecords: checkSimChanged enter
10-25 14:00:37.693  1361  1361 I SIMRecords: old iccid is 89014103786323167275
current is 89014503281383474393
```

Mobile Country Code Mobile Network Code (MCCMNC) (com.android.phone)

```
10-25 14:00:39.905  1361  1361 D SIMRecords: setVoiceMailByCountry: NetworkName
= 310410
10-25 14:00:39.905  1361  1361 D SIMRecords: getO2payState SIMState[READY]
MCCMNC[310410]
```

International Mobile Subscriber Identity (IMSI) (com.android.phone)

```
10-24 12:10:46.663  1362  1553 E GSMPhone:  Storing Voice Mail Count = 1 for
imsi = 312410635317425 for mVmCountKey = vm_count_key vmId = vm_id_key in
preferences.
```

Device Serial Number (/system/bin/at_distributor):

```
10-25 14:00:30.834   344   344 D ATD     : SERIALNO: R31D80L3M7Y
10-25 14:00:30.834   344   344 D ATD     : ril.serialnumber: R31D80L3M7Y
```

URLs opened via an Intent with the android.intent.action.VIEW action string (system_server)

```
10-25 18:44:03.695   808  1287 I am_create_activity:
[0,1131153416,58,com.android.chrome/com.google.android.apps.chrome.Main,android
.intent.action.VIEW,NULL,http://m.facebook.com/l.php?u=http://social.mgid.com/p
news/1935977/i/2384&h=BAQFWY6Tx&s=1&enc=AZOnerKuDJ68KZK4jEJxu2tIsnax1WLYlHp8O0p
sm4sU-y3u0DjDCg6bGhgdPDmDc4rTU4pbeTvnXilLojypSJXS,50331648,android]

10-25 18:44:20.962   808 12390 I am_create_activity:
[0,1163225456,58,android/com.android.internal.app.ResolverActivity,android.inte
nt.action.VIEW,NULL,http://m.facebook.com/l.php?u=http://www.answers.com/browse
/click.php?source=fb&param4=fb-us-mo-
gut&param3=www.answers.com%2Farticle%2F1235982%2F9-child-actors-you-wouldnt-
recognize-
today&param1=pop&param2=10455001&param5=10152232368716186&param6=10470511&h=4AQ
EHVjAz&s=1&enc=AZM0XenPF-
hKpDPI__vRxZqXcSfyyl_Xdzi5dZ5raZ4Vfk20qgGkUULBNZ8sVm_Yk5Rs5G2ia1JPWWPv3hFJQy9q,
8388608,com.facebook.katana]

10-25 18:45:00.761   808 12391 I am_create_activity:
[0,1123355696,58,android/com.android.internal.app.ResolverActivity,android.inte
nt.action.VIEW,NULL,http://m.facebook.com/l.php?u=http://www.richdadfreeseminar
.com/WashingtonDC/index.dtm?MID=5960769&h=4AQEHVjAz&s=1&enc=AZO31zAhAN9ihOhF9Gj
Bud72He8HQpwRBbp4fi-3bDzqJPhLWy8Gyxrx_GQOokcEF5-
zhsq1cUY8cEyV4wb2z7ct,8388608,com.facebook.katana]

10-25 18:45:35.995   808  1437 I am_create_activity:
[0,1165626376,58,android/com.android.internal.app.ResolverActivity,android.inte
nt.action.VIEW,NULL,http://m.facebook.com/l.php?u=http://slnm.us/AWfXNQ1&h=zAQF
qqEE_&s=1&enc=AZOCChHYxUwuquhQbH685LClYaH9eCo0ZiCDGEGfjSdxVRTn9s0VAv1k-
lFgTdFrjFK12___4WtiHcxfJgaYkciU,8388608,com.facebook.katana]
```

GPS coordinates opened via an Intent with the android.intent.action.VIEW action string (system_server)

```
10-25 18:45:50.279   808  1372 I am_create_activity:
[0,1154295608,58,android/com.android.internal.app.ResolverActivity,android.inte
nt.action.VIEW,NULL,geo:0,0?q=40.740758,-74.008382,8388608,com.facebook.katana]

10-25 18:44:50.501   808  1287 I am_create_activity:
[0,1132685568,58,android/com.android.internal.app.ResolverActivity,android.inte
nt.action.VIEW,NULL,geo:0,0?q=40.729701,-73.998511,8388608,com.facebook.katana]
```

Errors for URLs in the Chrome browser (com.android.chrome)

```
10-25 18:46:06.435 28281 28281 W chromium: [WARNING:data_provider.cc(48)] No
data for url http://travelocity.com/

10-25 18:46:06.435 28281 28281 W chromium: [WARNING:data_provider.cc(48)] No
data for url http://www.washingtonpost.com/world/national-security/us-releases-
images-it-says-show-russia-has-fired-artillery-over-border-into-
ukraine/2014/07/27/f9190158-159d-11e4-9e3b-7f2f110c6265_story.html

10-25 18:46:06.435 28281 28281 W chromium: [WARNING:data_provider.cc(48)] No
data for url
http://www.washingtonpost.com/blogs/worldviews/wp/2014/07/24/could-syrias-
islamist-fighters-hit-europe/

10-25 18:46:06.435 28281 28281 W chromium: [WARNING:data_provider.cc(48)] No
data for url http://m.washingtonpost.com/blogs/worldviews/wp/2014/07/24/could-
syrias-islamist-fighters-hit-europe/
```

```
10-25 18:46:06.435 28281 28281 W chromium: [WARNING:data_provider.cc(48)] No
data for url http://www.nbcnews.com/storyline/legal-pot/new-york-times-calls-
ending-federal-ban-marijuana-n165796

10-25 18:46:06.445 28281 28281 W chromium: [WARNING:data_provider.cc(48)] No
data for url http://www.huffingtonpost.com/2014/07/23/apple-ipad-
sales_n_5614474.html

10-25 18:46:06.535 28281 28281 W chromium: [WARNING:data_provider.cc(48)] No
data for url
https://www.travelocity.com/m/checkout?hotelId=40450&productKey=5a24067e-73b8-
488d-9690-
1826045389f7_V3&city=&sendEmailConfirmation=false&storeCreditCardInUserProfile=
false&checkInDate=2014-07-27&checkOutDate=2014-07-28&room1=2
```

Tell when the user is present and when they are not (system_server)

```
10-25 18:42:21.466   808   808 E MotionRecognitionService:
mReceiver.onReceive : ACTION_USER_PRESENT  :: UNLOCK SCREEN
```

Cell tower ID (com.android.phone):

```
10-24 12:11:11.157  1362  1362 D GsmSST  : RAT switched LTE:14 -> UMTS:3 at
cell 194212629
10-24 12:09:51.099  1362  1362 D GsmSST  : RAT switched UMTS:3 -> LTE:14 at
cell 167154192
```

The associated information for cell ID 194212629 is shown below (from http://opencellid.org/)
MCC: 310 (United States of America)
MNC: 410 (AT&T)
LAC: 10986
cell ID: 194212629

latitude: 38.852570
longitude: -77.300920

GPS raw data (system_server)

```
10-25 16:09:03.342   808  1551 E LocSvc_eng:   ephemeris mask: 856838770xn  almanac mask:
33125272
10-25 16:09:03.342   808  1551 E LocSvc_eng:   used in fix mask: 10000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:       sv: prn         snr       elevation
azimuth
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   0:   29    32.000000    56.000000
284.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   1:   2    0.000000    37.000000
102.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   2:   5    0.000000    58.000000
39.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   3:   6    0.000000    0.000000
0.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   4:   7    0.000000    2.000000
42.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   5:   10    0.000000    7.000000
60.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   6:   13    0.000000    12.000000
37.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   7:   15    0.000000    31.000000
188.000000
```

```
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   8:   18    0.000000    1.000000
248.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   9:   21    0.000000    10.000000
298.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   10:   25    0.000000    5.000000
233.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   11:   26    0.000000    53.000000
142.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   12:   30    0.000000    4.000000
71.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   13:   71    33.000000    73.000000
296.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   14:   70    0.000000    37.000000
32.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   15:   86    0.000000    75.000000
317.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   16:   79    0.000000    2.000000
29.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   17:   84    0.000000    0.000000
0.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   18:   87    0.000000    18.000000
327.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   19:   85    0.000000    46.000000
151.000000
10-25 16:09:03.342   808  1551 E LocSvc_eng:
10-25 16:09:03.342   808  1551 E LocSvc_eng: D/   20:   72    0.000000    28.000000
236.000000
```

Information about which applications are running on the device and when (system_server)

```
10-27 13:29:21.168   813  1447 I am_proc_start:
[0,8521,10117,com.sec.spp.push:RemoteNotiProcess,broadcast,com.sec.spp.push/.notisvc.regi
stration.PackageRemovedReceiver]
10-27 13:29:21.168   813  1474 I am_destroy_service: [0,1155223168,1341]
10-27 13:29:21.188   813  1320 I am_proc_bound:
[0,8521,com.sec.spp.push:RemoteNotiProcess]
10-27 13:29:21.278   813  2019 I am_proc_start:
[0,8534,10218,com.google.android.apps.docs,broadcast,com.google.android.apps.docs/.receiv
ers.AppPackageAddRemoveReceiver]
10-27 13:29:21.278   813  2019 I am_kill :
[0,7015,com.sec.android.widgetapp.SPlannerAppWidget,15,empty #31]
10-27 13:29:21.288   813  1398 I am_proc_died:
[0,7015,com.sec.android.widgetapp.SPlannerAppWidget,15]
10-27 13:29:21.298   813  1500 I am_proc_bound: [0,8534,com.google.android.apps.docs]
10-27 13:29:21.898   813  1474 I force_gc: Binder
10-27 13:29:22.128   813  1397 I am_create_service:
[0,1170837144,.ContentSyncService,10218,8534]
10-27 13:29:22.138   813   813 I notification_cancel: [com.android.vending,-
1050172287,NULL,0,0,64]
10-27 13:29:22.178   813  1162 I am_proc_start:
[0,8554,10044,com.sand.airdroid,broadcast,com.sand.airdroid/.components.apk.AppPackageRec
eiver]
10-27 13:29:22.189   813  1320 I am_proc_bound: [0,8554,com.sand.airdroid]
10-27 13:29:22.559   813   824 I am_proc_start:
[0,8584,10044,com.sand.airdroid:push,service,com.sand.airdroid/.servers.push.PushService]
10-27 13:29:22.589   813  1162 I am_proc_bound: [0,8584,com.sand.airdroid:push]
10-27 13:29:22.589   813  1162 I am_create_service:
[0,1166910520,.PushService,10044,8584]
```

```
10-27 13:29:22.649   813  1397 I am_proc_start:
[0,8596,10192,com.facebook.katana:dash,broadcast,com.facebook.katana/com.facebook.dash.re
ceivers.DashPackageUninstallReceiver]
10-27 13:29:22.659   813  1397 I am_kill : [0,6835,com.google.android.gm,15,empty #31]
10-27 13:29:22.699   813  1398 I am_proc_bound: [0,8596,com.facebook.katana:dash]
10-27 13:29:22.699   813  1320 I am_proc_died: [0,6835,com.google.android.gm,15]
```

MAC address (system_server)

```
10-24 17:34:21.014   812   920 D WifiDisplayController:  deviceAddress:
42:0e:85:7e:af:c9
```

Private IPv4 address (com.android.phone)

```
10-24 17:35:37.274  1353  1557 D DC-1    : REQ_GET_LINK_PROPERTIES
linkProperties{InterfaceName: rmnet_usb0 LinkAddresses: [10.137.74.110/32,]
Routes: [0.0.0.0/0 -> 10.137.74.109,] DnsAddresses: [172.26.38.1,172.26.38.2,]
Domains: nullMTU: 0}
```

IPv6 address (system_server)

```
10-27 14:04:18.085   813  1092 D ConnectivityService:    car=removed=[]
added=[fe80::420e:85ff:fe7e:afc9/64,]
```

Samsung Applications:

Name associated with a Gmail account (com.sec.android.app.FileShareServer)

```
10-25 15:30:38.523 30995 30995 D FileShare-Server:
ServerBroadcastReceiver.onReceive - All UserInfo: [UserInfo{0:Ray Bronson:13}]
```

Tell when the user is present and when they are not (com.samsung.android.app.gestureservice)

```
10-25 17:06:06.972  1558  1558 D GestureService: serviceOnOffReceiver:
onReceive - android.intent.action.USER_PRESENT
```

WiFi IPv4 address of device and default gateway in system properties from /data/log/dumpstate_app_native.txt.gz

```
[dhcp.wlan0.gateway]: [192.168.2.1]
[dhcp.wlan0.ipaddress]: [192.168.2.161]
[dhcp.wlan0.leasetime]: [86400]
[dhcp.wlan0.mask]: [255.255.255.0]
```

The MAC address of the device (400e85fffe7eafc9) and IPv6 address (fe80000000000000420e85fffe7eafc9) from /data/log/dumpstate_app_native.txt.gz

```
------ NETWORK ROUTES IPV6 (/proc/net/ipv6_route) ------
00000000000000000000000000000000 00 00000000000000000000000000000000 00
00000000000000000000000000000000 ffffffff 00000001 00000382 00200200       lo
fe800000000000000000000000000000 40 00000000000000000000000000000000 00
00000000000000000000000000000000 00000100 00000000 00000000 00000001     p2p0
fe800000000000000000000000000000 40 00000000000000000000000000000000 00
00000000000000000000000000000000 00000100 00000000 00000000 00000001    wlan0
00000000000000000000000000000000 00 00000000000000000000000000000000 00
00000000000000000000000000000000 ffffffff 00000001 00000382 00200200       lo
00000000000000000000000000000001 80 00000000000000000000000000000000 00
00000000000000000000000000000000 00000000 00000001 00000000 80200001       lo
fe8000000000000400e85fffe7eafc9 80 00000000000000000000000000000000 00
00000000000000000000000000000000 00000000 00000001 00000000 80200001       lo
fe8000000000000420e85fffe7eafc9 80 00000000000000000000000000000000 00
00000000000000000000000000000000 00000000 00000001 00000000 80200001       lo
ff000000000000000000000000000000 08 00000000000000000000000000000000 00
00000000000000000000000000000000 00000100 00000000 00000000 00000001     p2p0
ff000000000000000000000000000000 08 00000000000000000000000000000000 00
00000000000000000000000000000000 00000100 00000000 00000000 00000001    wlan0
00000000000000000000000000000000 00 00000000000000000000000000000000 00
00000000000000000000000000000000 ffffffff 00000001 00000382 00200200       lo
```

Appendix B: Android log messages from the com.healthagen.iTriage Android application

User Login (com.healthagen.iTriage)

```
10-27 13:05:39.785  7157  7157 D MARK    : appboy login stuff (my itriage
login): 1237397, feelsgoodman@gmail.com
```

User Password (com.healthagen.iTriage)

```
10-27 13:05:36.392  7157  7206 D JOSH    : key 1 generated from password:
*trap_Door[190*
```

Cookie (com.healthagen.iTriage)

```
10-27 12:31:29.055 29172 29172 D j       : Cookie:
_itriage_session_tracker=f1ebcf48939fb7968ebe962ff6c46b55.1414426817;
_itriage_unique_tracker=75a4db35d1bd977ddd3cff3acfe9fc3d.1414426817;
_itriage_session=8aa8c6bf4d5be5f2c83ab523e77130355c4fc3f575da31fef08d8a458f5d4e
4bbd5e93c9d65c1463a3cb59541df5550e02148f5ad2f2256094b2f89412f6d0fb;
env=production
10-27 12:31:29.055 29172 29172 D j       : Cookie2: $Version=1
10-27 12:31:29.055 29172 29172 D j       : X-CSRF-Token:
YmawWVNBGFFvojUTev65opDRme7UB20KFq3r2iScjAs=
```

Security Token (com.healthagen.iTriage)

```
10-27 12:30:00.308 29172 29203 I REQUEST PAYLOAD:
https://healthnews.itriagehealth.com/preferred_articles.json?installation_id=e6
608bc8-f652-425c-b02e-
512e59b665fb&security_token=704e16fc5d724657e34d1665c83ac39a2da5c121&page=1&per
_page=25&last_updated_since=1414165392.879
```

Medical Procedures (com.healthagen.iTriage)

```
10-27 12:30:49.766 29172 29172 D MARK    :
{"id":340,"item_note":"Finally","procedure_date":"10\/27\/2014","_etag":"","pro
cedure_doctor":"Dr. Robot","_deleted":false,"name":"Buttocks lift"}
```

Medications (com.healthagen.iTriage)

```
10-27 12:29:55.613 29172 29172 D DUCK    : {"4d754d1b-dbbe-4d04-8df9-
61df080e130f":{"id":609,"medication_doctor":"Dr.
Robot","name":"Diazepam","item_note":"","medication_date":"10\/27\/2014","_awai
tingUpload":true,"_deleted":false,"dosages":[{"dose_type_id":3,"amount":"100mg"
,"method_type_id":5,"frequency":"All the time"}]}}
```

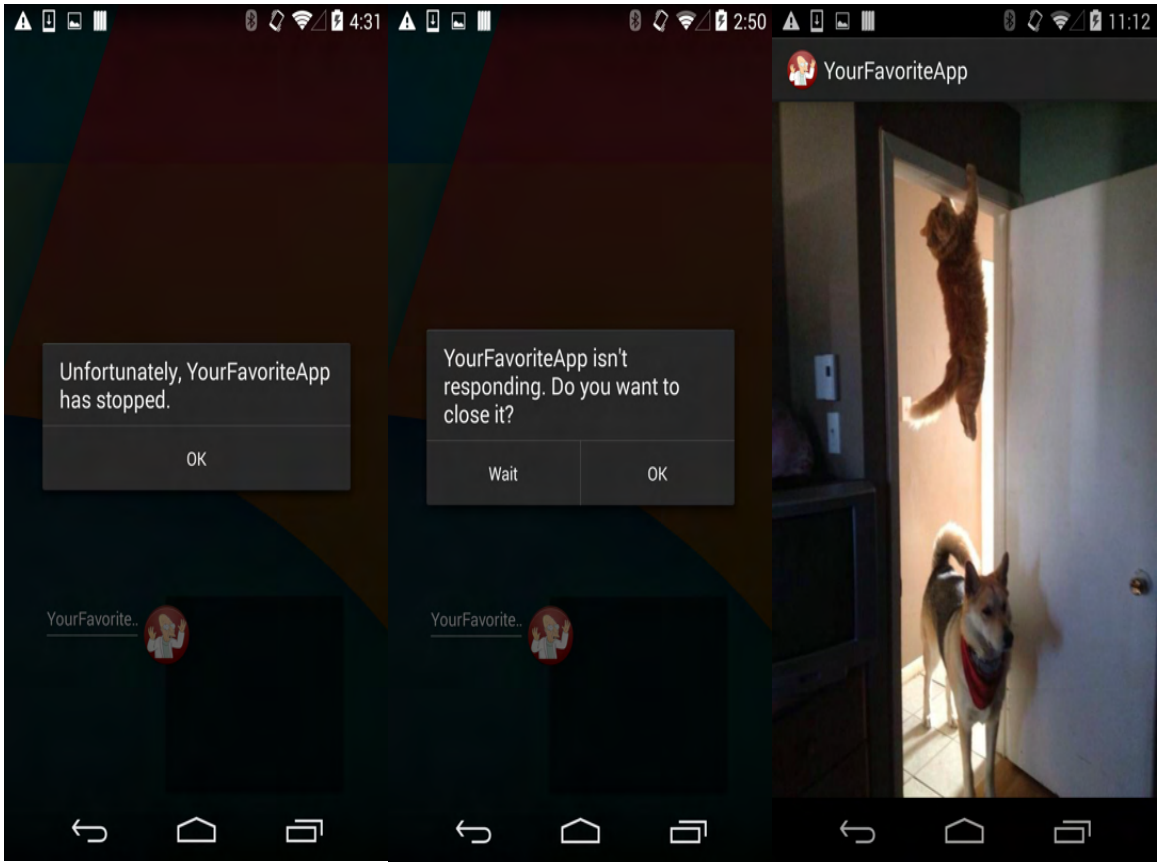
Conditions (com.healthagen.iTriage)

```
10-27 13:06:48.082  7157  7157 D DUCK    : {"0797760b-433b-494d-b5d3-
a5b71909b931":{"id":10,"item_note":"O_o","_awaitingUpload":true,"_deleted":fals
e,"disease_date":"10\/27\/2014","disease_doctor":"Dr. Robot","name":"Acid (LSD)
abuse"}}
```

Insurance Information (Aetna Group Number W3342212105 Member ID: 1234567-123-12345) (com.healthagen.iTriage)

```
10-27 14:20:54.417 18900 18993 V z       :
https://www.itriagehealth.com/api/v1/narrow_network/33/validate_member_info/W33
42212105/1234567-231-12345 -> HTTP/1.1 401 Unauthorized
```

Appendix C: Screenshots, in order from left to right, of an uncaught exception, Application Not Responding, and native code error



Appendix D: Snippet of android.app.Notification.toString() method in smali format from a Samsung Android 4.4.2 build

```
    const-string v3, " contentTitle="

    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    .line 1337
    iget-object v3, p0, Landroid/app/Notification;->contentTitle:Ljava/lang/CharSequence;

    if-eqz v3, :cond_186

    .line 1338
    iget-object v3, p0, Landroid/app/Notification;->contentTitle:Ljava/lang/CharSequence;

    invoke-virtual {v3}, Ljava/lang/Object;->toString()Ljava/lang/String;

    move-result-object v3

    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    .line 1342
    :goto_c3
    const-string v3, " contentText="

    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    .line 1343
    iget-object v3, p0, Landroid/app/Notification;->contentText:Ljava/lang/CharSequence;

    if-eqz v3, :cond_18d

    .line 1344
    iget-object v3, p0, Landroid/app/Notification;->contentText:Ljava/lang/CharSequence;

    invoke-virtual {v3}, Ljava/lang/Object;->toString()Ljava/lang/String;

    move-result-object v3

    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    .line 1348
    :goto_d5
    const-string v3, " tickerText="

    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    .line 1349
    iget-object v3, p0, Landroid/app/Notification;->tickerText:Ljava/lang/CharSequence;

    if-eqz v3, :cond_194

    .line 1350
    iget-object v3, p0, Landroid/app/Notification;->tickerText:Ljava/lang/CharSequence;

    invoke-virtual {v3}, Ljava/lang/Object;->toString()Ljava/lang/String;

    move-result-object v3

    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    .line 1354
    :goto_e7
    const-string v3, " kind=["
```

```
    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    .line 1355
    iget-object v3, p0, Landroid/app/Notification;->kind:[Ljava/lang/String;

    if-nez v3, :cond_19b

    .line 1356
    const-string v3, "null"

    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

    .line 1363
    :cond_f5
    const-string v3, "]"

    invoke-virtual {v2, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;
```

Appendix E: Anonymized Android Notifications as They Appear in the Android Log

<Data obtained> <process name that wrote the log message>
<Android log message(s) of the format (timestamp, PID, PPID, Log Level, Log Tag, Log Message)>

Google Chat (com.google.android.talk)

```
10-23 16:51:21.555   810   821 I notification_enqueue:
[com.google.android.talk,0,com.google.android.talk:chat:1,0,Notification(pri=0
icon=7f02056d contentView=com.google.android.talk/0x1090086 vibrate=null
sound=null defaults=0x4 flags=0x11 when=1414097480631 ledARGB=0x0
contentIntent=Y deleteIntent=Y contentTitle=Mohammed Al Amri contentText=i
believe the path is invalid somehow tickerText=Mohammed Al Amri: i believe the
path is invalid somehow kind=[null])]

10-23 17:11:37.982   810  1443 I notification_enqueue:
[com.google.android.talk,0,com.google.android.talk:chat:1,0,Notification(pri=0
icon=7f02056d contentView=com.google.android.talk/0x1090086 vibrate=default
sound=android.resource://com.google.android.talk/raw/2131230729 defaults=0x6
flags=0x11 when=1414098697414 ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=Mohammed Al Amri contentText=it seems that "unity" game engine
thing only works in c# tickerText=Mohammed Al Amri: it seems that "unity" game
engine thing only works in c# kind=[null])]

10-24 12:46:43.878   810  1451 I notification_enqueue:
[com.google.android.talk,0,com.google.android.talk:chat:1,0,Notification(pri=0
icon=7f020573 contentView=com.google.android.talk/0x1090086 vibrate=default
sound=android.resource://com.google.android.talk/raw/2131230729 defaults=0x6
flags=0x11 when=1414169204032 ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=Vanderlei Silva contentText=Korax tickerText=Vanderlei Silva:
Korax kind=[null])]
```

Gmail (com.google.android.gm)

```
10-23 16:25:19.191   810  1443 I notification_enqueue: [com.google.android.gm,-
1847466507,NULL,0,Notification(pri=0 icon=7f0200f6
contentView=com.google.android.gm/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x19
when=1414095918915 ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=WSJ.com Editors contentText=WSJ NEWS ALERT: Amazon Spending Leads
to Another Loss tickerText=WSJ.com Editors kind=[null] 2 actions)]
10-23 16:25:19.201   810   821 I am_destroy_service: [0,1160015232,32230]

10-23 17:20:09.991   810  1467 I notification_enqueue: [com.google.android.gm,-
1847466507,NULL,0,Notification(pri=0 icon=7f0200f6
contentView=com.google.android.gm/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x11
when=1414099209708 ledARGB=0x0 contentIntent=Y deleteIntent=Y contentTitle=Manu
Rama Dev contentText=Are we signed up for the demo room? tickerText=Manu Rama
Dev kind=[null] 2 actions)]

10-23 18:18:51.310   810  1415 I notification_enqueue: [com.google.android.gm,-
1847466507,NULL,0,Notification(pri=0 icon=7f0200f6
contentView=com.google.android.gm/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x11
when=1414102730962 ledARGB=0x0 contentIntent=Y deleteIntent=Y contentTitle=Ray
Bronson contentText=Generic Subject tickerText=Ray Bronson kind=[null] 2
actions)]
```

Google Email Client (com.google.android.email)

```
10-28 16:06:23.465   813  1417 I notification_enqueue:
[com.android.email,268435458,NULL,0,Notification(pri=0 icon=7f02029c
contentView=com.android.email/0x1090086 vibrate=null sound=null defaults=0x4
flags=0x1 when=1414526783429 ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=Ray Bronson contentText=Long Subject Long Subject Long Subject
Long Subject Long Subject Long Subject Long Subject Long Subject tickerText=N
kind=[null])]

10-28 15:58:58.371   813 10600 I notification_enqueue:
[com.android.email,268435458,NULL,0,Notification(pri=0 icon=7f02029c
contentView=com.android.email/0x1090086 vibrate=null sound=null defaults=0x4
flags=0x1 when=1414526338279 ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=Ray Bronson contentText=Hello tickerText=N kind=[null])]
```

Facebook Messenger (com.facebook.orca)

```
10-23 17:36:45.803   810  1391 I notification_enqueue:
[com.facebook.orca,10000,t_mid.1414042734100:816f873c7b5c15b730,0,Notification(
pri=1 icon=7f020795 contentView=com.facebook.orca/0x1090086 vibrate=null
sound=null defaults=0x0 flags=0x1 when=1414100204900 ledARGB=0xff00ff00
contentIntent=Y deleteIntent=N contentTitle=Qin Yi contentText=There is no
credits in my classes tickerText=Qin Yi: There is no credits in my classes
kind=[null] 1 action)]

10-27 11:32:29.021   812  1110 I notification_enqueue:
[com.facebook.orca,10000,t_mid.1409980055603:fdb4f1c34c667a2e61,0,Notification(
pri=1 icon=7f02079e contentView=com.facebook.orca/0x1090086 vibrate=null
sound=null defaults=0x0 flags=0x1 when=1414423947105 ledARGB=0xff00ff00
contentIntent=Y deleteIntent=N contentTitle=EunJoo  Kim contentText=すみません!!!
最近めっちゃ忙しかったです。。また用事とかいろいろあっていつ会えるかはまだわかりません T.T また
連絡します~ tickerText=EunJoo: すみません!!!  最近めっちゃ忙しかったです。。また用事とかい
ろいろあっていつ会えるかはまだわかりません T.T また連絡しま… kind=[null] 1 action)]

10-18 20:41:41.553   804  1450 I notification_enqueue:
[com.facebook.orca,10000,t_mid.1409273689993:bc731045d56f2b9465,0,Notification(
pri=1 icon=7f020795 contentView=com.facebook.orca/0x1090086 vibrate=null
sound=null defaults=0x0 flags=0x1 when=1413666578297 ledARGB=0xff00ff00
contentIntent=Y deleteIntent=N contentTitle=Newsha Hashemi contentText=That's
great tickerText=Newsha: That's great kind=[null] 1 action)]

10-18 20:41:43.574   804  1435 I notification_enqueue:
[com.facebook.orca,10000,t_msg.b9760579fecd036cfacb18e022c97c1431,0,Notificatio
n(pri=1 icon=7f020795 contentView=com.facebook.orca/0x1090086 vibrate=null
sound=null defaults=0x0 flags=0x1 when=1413362544722 ledARGB=0xff00ff00
contentIntent=Y deleteIntent=N contentTitle=Mindy Mindi contentText=no in
mongolia tickerText=Mindy: no in mongolia kind=[null] 1 action)]
```

Missed Calls (com.android.phone)

```
10-23 18:14:19.862   810   820 I notification_enqueue:
[com.android.phone,1,NULL,0,Notification(pri=2 icon=108007f
contentView=com.android.phone/0x1090086 vibrate=null sound=null defaults=0x4
flags=0x11 when=1414102438778 ledARGB=0x0 contentIntent=Y deleteIntent=N
contentTitle=Missed call contentText=(703) 537-0616 tickerText=Missed call from
(703) 838-0616 kind=[null] 2 actions)]
```

Text Messages (com.android.mms)

```
10-23 18:16:08.911   810  1466 I notification_enqueue:
[com.android.mms,123,NULL,0,Notification(pri=2 icon=7f0202c3
contentView=com.android.mms/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x1
when=1414102566000 ledARGB=0x0 contentIntent=Y deleteIntent=Y
```

```
contentTitle=Leanna contentText=heyyy tickerText=Leanna: heyyy
kind=[android.message])]

10-23 18:16:38.170   810  1466 I notification_enqueue:
[com.android.mms,123,NULL,0,Notification(pri=2 icon=7f0202c3
contentView=com.android.mms/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x1
when=1414102596000 ledARGB=0x0 contentIntent=Y deleteIntent=Y contentTitle=Fred
M Smith contentText=You bet. tickerText=Fred M Smith: You bet.
kind=[android.message])]
```

Google Password Reset via Text Message (com.android.mms)

```
10-25 15:27:09.840   808  1410 I notification_enqueue:
[com.android.mms,123,NULL,0,Notification(pri=2 icon=7f0202c3
contentView=com.android.mms/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x1
when=1414265227000 ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=224444 contentText=Your Google verification code is 609483
tickerText=224444: Your Google verification code is 609483
kind=[android.message])]

10-25 16:55:59.980   808  1453 I notification_enqueue:
[com.android.mms,123,NULL,0,Notification(pri=2 icon=7f0202c3
contentView=com.android.mms/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x1
when=1414270557000 ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=+18133360555 contentText=Your Google verification code is 390984
tickerText=+18133360555: Your Google verification code is 390984
kind=[android.message])]
```

Facebook Password Reset Via Text Message (com.android.mms)

```
10-25 16:57:42.680   808  1452 I notification_enqueue:
[com.android.mms,123,NULL,0,Notification(pri=2 icon=7f0202c3
contentView=com.android.mms/0x1090086 vibrate=null
sound=content://settings/system/notification_sound defaults=0x4 flags=0x1
when=1414270660000 ledARGB=0x0 contentIntent=Y deleteIntent=Y
contentTitle=32665 contentText=597872 is your Facebook Password reset code or
reset your password here: https://fb.com/l/1NQ06FJpVVNR87Q  tickerText=32665:
597872 is your Facebook Password reset code or reset your password here:
https://fb.com/l/1NQ06FJpVVNR87Q  kind=[android.message])]
```

Google Maps Directions (com.google.android.apps.maps)

```
10-24 12:51:59.126   810  1379 I notification_enqueue:
[com.google.android.apps.maps,39796916,NULL,0,Notification(pri=1 icon=7f02023b
contentView=com.google.android.apps.maps/0x1090086 vibrate=null sound=null
defaults=0x0 flags=0x2 when=1414169519075 ledARGB=0x0 contentIntent=Y
deleteIntent=N contentTitle=Korax Inc contentText=Head west toward University
Dr tickerText=N kind=[null] 1 action)]
```

Whatsapp Messages (com.whatsapp)

```
10-24 12:30:01.440   810  1387 I notification_enqueue:
[com.whatsapp,1,NULL,0,Notification(pri=0 icon=7f0205c2
contentView=com.whatsapp/0x1090086 vibrate=default sound=null defaults=0x2
flags=0x1 when=1414168201281 ledARGB=0xffffffff contentIntent=Y deleteIntent=N
contentTitle=Manu contentText=Lol. tickerText=Message from Manu kind=[null])]
```

Connection to WiFi (com.android.settings)

```
10-20 23:23:31.855   810  1404 I notification_enqueue:
[com.android.settings,3012971,NULL,0,Notification(pri=-2 icon=10807c3
contentView=com.android.settings/0x1090086 vibrate=null sound=null defaults=0x0
flags=0x2 when=0 ledARGB=0x0 contentIntent=Y deleteIntent=N contentTitle=Wi-Fi
```

```
connected contentText=Connected to "NETGEAR242". tickerText=Wi-Fi connected
kind=[null])]

10-24 15:57:42.790   811  1410 I notification_enqueue:
[com.android.settings,3012971,NULL,0,Notification(pri=-2 icon=10807c3
contentView=com.android.settings/0x1090086 vibrate=null sound=null defaults=0x0
flags=0x2 when=0 ledARGB=0x0 contentIntent=Y deleteIntent=N contentTitle=Wi-Fi
connected contentText=Connected to "korax-5GHz". tickerText=Wi-Fi connected
kind=[null])]
```

AirDroid (com.sand.airdroid)

```
10-25 14:34:28.315   808   934 I notification_enqueue:
[com.sand.airdroid,101,NULL,0,Notification(pri=0 icon=7f020160
contentView=com.sand.airdroid/0x1090086 vibrate=null sound=default defaults=0x1
flags=0x4a when=1414262068320 ledARGB=0x0 contentIntent=Y deleteIntent=N
contentTitle=AirDroid contentText=Connected from: 192.168.1.12 tickerText=N
kind=[null])]

10-23 17:49:09.759   810   919 I notification_enqueue:
[com.sand.airdroid,101,NULL,0,Notification(pri=0 icon=7f020160
contentView=com.sand.airdroid/0x1090086 vibrate=null sound=default defaults=0x1
flags=0x4a when=1414100949763 ledARGB=0x0 contentIntent=Y deleteIntent=N
contentTitle=AirDroid contentText=Service is running tickerText=N kind=[null])]
```

Random Notifications (Various Applications)

```
10-28 15:49:52.578   813   824 I notification_enqueue:
[com.google.android.googlequicksearchbox,6,NULL,0,Notification(pri=-2
icon=7f02030e contentView=com.google.android.googlequicksearchbox/0x1090086
vibrate=null sound=null defaults=0x0 flags=0x18 when=1414525792540 ledARGB=0x0
contentIntent=Y deleteIntent=Y contentTitle=Time to Home contentText=ETA: 30
min (3 minutes delay) via US-50 W and VA-28 N tickerText=N kind=[null] 1
action)]

10-25 15:26:05.917   808  1387 I notification_enqueue:
[com.google.android.gms,1,NULL,0,Notification(pri=0 icon=108008a
contentView=com.google.android.gms/0x1090086 vibrate=null sound=null
defaults=0x0 flags=0x10 when=1414265165923 ledARGB=0x0 contentIntent=Y
deleteIntent=N contentTitle=Sign-in Request contentText=whymewhynot@gmail.com
tickerText=Access Requested kind=[null])]

10-25 15:30:24.179   808  1436 I notification_enqueue:
[com.google.android.gms,3,NULL,0,Notification(pri=0 icon=108008a
contentView=com.google.android.gms/0x1090086 vibrate=null sound=null
defaults=0x0 flags=0x10 when=1414265424182 ledARGB=0x0 contentIntent=Y
deleteIntent=N contentTitle=Sign-in Request
contentText=toomuchisnotenough@gmail.com tickerText=Access Requested
kind=[null])]

10-27 12:12:15.359   812  1449 I notification_enqueue:
[com.google.android.googlequicksearchbox,6,NULL,0,Notification(pri=-2
icon=7f020311 contentView=com.google.android.googlequicksearchbox/0x1090086
vibrate=null sound=null defaults=0x0 flags=0x18 when=1414426335354 ledARGB=0x0
contentIntent=Y deleteIntent=Y contentTitle=61° - Partly Cloudy contentText=N
tickerText=N kind=[null])]

10-27 13:28:56.433   813  1446 I notification_enqueue: [com.android.vending,-
1050172287,NULL,0,Notification(pri=-1 icon=1080081
contentView=com.android.vending/0x1090086 vibrate=null sound=null defaults=0x0
flags=0x10 when=1414430936444 ledARGB=0x0 contentIntent=Y deleteIntent=N
contentTitle=Candy Crush Saga contentText=Updating "Candy Crush Saga"…
tickerText=Updating "Candy Crush Saga"… kind=[null])]
```

10-27 12:19:46.069   812   822 I notification_enqueue: [com.android.vending,-1050172287,NULL,0,Notification(pri=-1 icon=1080081 contentView=com.android.vending/0x1090086 vibrate=null sound=null defaults=0x0 flags=0x10 when=1414426786068 ledARGB=0x0 contentIntent=Y deleteIntent=N contentTitle=iTriage Health contentText=Installing "iTriage Health"… tickerText=Installing "iTriage Health"… kind=[null])]

10-23 17:47:44.706   810 1390 I notification_enqueue: [android,17041217,NULL,-1,Notification(pri=0 icon=1080464 contentView=android/0x1090086 vibrate=null sound=null defaults=0x0 flags=0x1002 when=0 ledARGB=0x0 contentIntent=Y deleteIntent=N contentTitle=Choose input method contentText=Samsung keyboard tickerText=N kind=[android.system.imeswitcher])]

10-18 20:40:57.414   804 1451 I notification_enqueue: [com.android.systemui,273,NULL,0,Notification(pri=0 icon=7f020141 contentView=com.android.systemui/0x1090086 vibrate=null sound=null defaults=0x0 flags=0x2 when=0 ledARGB=0x0 contentIntent=N deleteIntent=N contentTitle=No SIM contentText=Insert SIM card tickerText=No SIM kind=[null])]

10-18 20:41:39.270   804 1451 I notification_enqueue: [com.kiloo.subwaysurf,0,NULL,0,Notification(pri=0 icon=7f020040 contentView=com.kiloo.subwaysurf/0x1090086 vibrate=[100,250,100,500] sound=android.resource://com.kiloo.subwaysurf/raw/cpush defaults=0x0 flags=0x10 when=1413679299242 ledARGB=0x0 contentIntent=Y deleteIntent=N contentTitle=Subway Surfers contentText=Cash in on the Sale this Weekend! Buy a Special Deal pack and get 20,000 extra Coins! tickerText=Cash in on the Sale this Weekend! Buy a Special Deal pack and get 20,000 extra Coins! kind=[null])]

10-18 20:41:48.269   804 1434 I notification_enqueue: [com.android.vending,-234430262,NULL,0,Notification(pri=-1 icon=7f02018b contentView=com.android.vending/0x1090086 vibrate=null sound=null defaults=0x0 flags=0x10 when=1413679308259 ledARGB=0x0 contentIntent=Y deleteIntent=N contentTitle=5 updates available contentText=Touch to update YP - Yellow Pages local search, myAT&T, GROUP PLAY, ChatON, and Doodle Jump. tickerText=5 updates available kind=[null])]

References

[1] READ_LOGS permission is not granted to 3rd party applications in Jelly Bean (api 16). https://groups.google.com/forum/#!msg/android-developers/6U4A5irWang/9Uj7h5CLTgcJ

[2] Manifest.permission | Android Developers. http://developer.android.com/reference/android/Manifest.permission.html#READ_LOGS

[3] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang, "AppIntent: Analyzing Sensitive Data Transmission in Android for Privacy Leakage Detection," in Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS), 2013.

[4] cmds/dumpstate/dumpstate.c - platform/frameworks/native - Git at Google. https://android.googlesource.com/platform/frameworks/native/+/master/cmds/dumpstate/dumpstate.c

[5] Keeping Your App Responsive | Android Developers. http://developer.android.com/training/articles/perf-anr.html

[6] Unix signal - Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Unix_signal

[7] Bypassing Android Permissions: What You Need to Know. http://blog.trendmicro.com/trendlabs-security-intelligence/bypassing-android-permissions-what-you-need-to-know/

[8] GC: Notification - android.app.Notification (.java) - GrepCode Class Source. http://grepcode.com/file/repository.grepcode.com/java/ext/com.google.android/android/4.4.2_r1/android/app/Notification.java#Notification

[9] smali - An assembler/disassembler for Android's dex format - Google Project Hosting. https://code.google.com/p/smali/

[10] services/java/com/android/server/NotificationManagerService.java - platform/frameworks/base.git - Git at Google . https://android.googlesource.com/platform/frameworks/base.git/+/android-4.4.2_r1/services/java/com/android/server/NotificationManagerService.java

[11] logcat/event-log-tags - platform/system/core - Git at Google. https://android.googlesource.com/platform/system/core/+/b084929f5dd57b878f6debe6567a6c8888061fa0/logcat/event-log-tags