# ANALISIS FORENSIK MALWARE PADA PLATFORM ANDROID

**Rahmat Novrianda[1)], Yesi Novaria Kunang[2)], P.H. Shaksono[3)]**

*[1),3)] Magister Teknik Informatika, Universitas Bina Darma*
*[2)] Program Studi Sistem Informasi, Universitas Bina Darma*
*Jl. Ahmad Yani no. 3 Plaju Palembang*
*Email :* rahmatnovri.android19@gmail.com[1)], yesi_kunang@mail.binadarma.ac.id[2)]

**Abstrak**

*Penggunaan smartphone berplatform android sebagai sistem operasi open source sangat marak. Akan tetapi karena android merupakan sistem open source memudahkan siapa saja untuk mengembangkan aplikasi android yang bisa di download di Android App Market. Termasuk aplikasi yang disisipi malware oleh pengembang aplikasi.*

*Pada penelitian ini mengambil sample tiga aplikasi android yang mengandung malware aplikasi android iCalendar (Android.raden.A), Live prints live wallpaper (Hongtoutou) dan Hippo (Hippo SMS). Selanjutnya, dilakukan analisa terhadap ketiga aplikasi android tersebut dengan menggunakan teknik komputer forensik. Penelitian ini sangat berguna agar user dapat menganalisa aplikasi android lainnya dan menghindari masuknya malware android ke perangkat android milik user. Tahapan analisis ini dimulai dengan merename apk menjadi zip, kemudian mengekstark file zip tadi sehingga menjadi format DEX. File DEX tersebut dikonvert menjadi file JAR dengan tool Dex2jar. Kemudian hasil file JAR didecompile dengan JD-GUI untuk melihat source code java yang kemudian dianalisis.*

*Penelitian ini memberikan gambaran cara kerja ketiga malware android tersebut pada sistem android dan dampak yang diakibatkannya terhadap sistem android. Hasil analisis memperlihatkan malware android bekerja antara lain dengan mencuri informasi berupa IMEI dan IMSI dari perangkat, selain itu juga sebagian malware bekerja mengirimkan SMS ke premium number yang sangat merugikan pemilik perangkat.*

***Kata kunci:*** *malware, android, komputer forensik, iCalender, live prints live wallpaper, hippo*

## 1. Pendahuluan

### 1.1. Latar Belakang

*Malware* merupakan sebuah *software* yang dapat menyusup ke sistem operasi sehingga dapat merusak sistem dan juga dapat mencuri file-file penting yang ada pada sistem. *Malware* atau juga disebut software perusak mencakup Virus Komputer, *Trojan Horse*, perangkat pengintai (*spyware*), perangkat iklan (*adware*) yang tidak jujur, perangkat jahat (*crimeware*) dan perangkat lunak lainnya yang berniat jahat dan tidak diinginkan. Berkembangnya teknologi pun memicu dikembangkannya *malware-malware* baru, sehingga semakin banyak pihak yang dirugikan karena adanya *malware-malware* ini. Bahkan saat ini, *malware* telah dapat menjangkit hampir seluruh jenis sistem operasi.

Dengan kemajuan teknologi saat ini yang mulai meluncurkan berbagai jenis *gadget* seperti *tablet PC* dan *smartphone*, tentunya didukung oleh beberapa jenis sistem operasi. Semakin maraknya penggunaan *tablet PC* dan *smartphone* oleh user, dikarenakan ukurannya yang gampang dibawa dan juga tidak terlalu sulit untuk mengoperasikannya. Sejalan dengan hal tersebut, salah satu sistem operasi yang sedang marak digunakan untuk mendukung kerja *tablet PC* ataupun *smartphone* adalah sistem android.

Sistem android adalah sistem operasi yang berbasis linux untuk telepon seluler seperti *smartphone* dan *tablet PC*. Sistem android memiliki keunggulan, seperti sistem operasi bersifat *open source*, *multitasking*, kemudahan dalam notifikasi hingga banyaknya aplikasi atau *software* yang dapat dinikimati dengan menggunakan sistem android. Akan tetapi, salah satu keunggulan sistem android menjadi salah satu kelemahannya. Keunggulan sistem operasi bersifat *open access*, dimana disediakan *platform* terbuka bagi para pengembang (*user*) yang dimaksudkan agar *user* dapat menciptakan dan mengembangkan aplikasi mereka sendiri sehingga dapat digunakan pada bermacam perangkat seluler. Akan tetapi, hal ini malah menimbulkan kemudahan dalam pihak yang tidak bertanggung jawab untuk membangun dan mengembangkan *malware* menjadi aplikasi yang dapat masuk ke sistem android.

Beberapa perusahaan antivirus yang mengeluarkan aplikasi antivirus untuk platform Android mengemukakan sebuah penemuan tentang *malware* android. Mereka telah merekap semua data yang mereka peroleh dari aplikasi produk antivirus mereka yang terpasang pada perangkat Android di 118 negara di dunia [6]. Dari hasil rekapitulasi yang dilakukan ditemukanlah beberapa *malware* android yang sering menyerang perangkat android, diantaranya adalah *Android.raden.A, Hongtoutou dan Hippo SMS*.

Pada penelitian ini akan mengambil sample tiga aplikasi android yang mengandung ketiga *malware* android yang telah disebutkan di atas, dimana aplikasi android

*iCalendar (Android.raden.A), Live prints live wallpaper (Hongtoutou)* dan *Hippo (Hippo SMS)*. Selanjutnya, dilakukan analisa terhadap ketiga aplikasi android tersebut dengan menggunakan teknik komputer forensik. Penelitian ini dianggap sangat berguna agar user dapat menganalisa aplikasi android lainnya dan menghindari masuknya *malware* android ke perangkat android milik user. Selain itu, hasil penelitian ini memberikan gambaran cara kerja ketiga *malware* android tersebut pada sistem android dan dampak yang diakibatkannya terhadap sistem android.

**1.2. Komputer Forensik**

Komputer Forensik bertujuan untuk mengamankan dan menganalisa bukti digital. Dari data yang diperoleh melalui survei oleh FBI dan *The Computer Security Institute* pada tahun 1999, mengatakan bahwa 51% responden mengakui bahwa mereka telah menderita kerugian, terutama dalam bidang finansial akibat kejahatan komputer [2].

Komputer forensik hanyalah penerapan teknik komputer investigasi dan analisis untuk kepentingan menentukan bukti hukum potensial. Bukti mungkin dapat dicari dalam berbagai kejahatan komputer atau penyalahgunaan, namun tidak terbatas pada pencurian rahasia perdagangan, pencurian atau perusakan kekayaan intelektual serta penipuan. Seorang CHFI dapat menarik berbagai metode untuk menemukan data yang berada dalam sistem komputer, atau memulihkan yang telah dihapus, informasi file terenkripsi atau rusak.

Komputer forensik juga banyak sekali digunakan untuk tugas-tugas lainya seperti : *Operational Troubleshooting*, *Log Monitoring*, *Data Recovery* dan *Data Acquisition* yaitu mengambil data dari *host* dan *Due Diligence (Regulatory Compliance)* untuk kepentingan audit.

Komputer Forensik memiliki Prinsip Dasar yang harus diterapkan, yaitu : (a) Forensik bukan proses *Hacking*; (b) Data yang didapat harus dijaga agar jangan berubah.; (c) Penting untuk Membuat image dari *HD/Floppy/ USB-Stick/Memory-Dump* tanpa merubah isinya, yang terkadang digunakan hardware khusus.; (d) *Image* yang dianalisis bukan yang asli.; (e) Data yang sudah terhapus membutuhkan *tools* khusus untuk merekonstruksi.; (f) Pencarian bukti menggunakan tools pencarian teks khusus, atau mencari satu persatu dalam image.

Berikut prosedur forensik menurut metode *Search* dan *Seizure* adalah: (a) Identifikasi dan penelitian permasalahan.; (b). Membuat hipotesa.; (c) Uji hipotesa secara konsep dan empiris.; (d) Evaluasi hipotesa berdasarkan hasil pengujian dan pengujian ulang jika hipotesa tersebut jauh dari apa yang diharapkan.; (e) Evaluasi hipotesa terhadap dampak yang lain jika hipotesa tersebut dapat diterima [7].

**1.3. *Malware***

*Malware* berasal dari kata *malicious* dan *software* yang merupakan perangkat lunak yang diciptakan untuk menyusup atau merusak sistem komputer. Istilah ini umum dipakai oleh pakar komputer untuk mengartikan berbagai macam perangkat lunak atau kode perangkat lunak yang mengganggu atau mengusik.

Perangkat lunak yang dianggap sebagai perangkat perusak berdasarkan maksud yang terlihat dari pencipta dan bukan berdasarkan ciri-ciri tertentu, mencakup *Virus Computer, Trojan Horse,* perangkat pengintai (*spyware)*, perangkat iklan (*adware*) yang tidak jujur, perangkat jahat (*crimeware*) dan perangkat lunak lainnya yang berniat jahat dan tidak diinginkan [3].

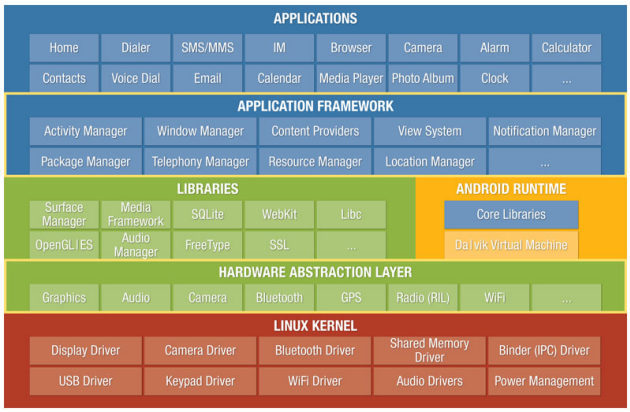
Perangkat perusak tidak sama dengan perangkat lunak cacat (*defective software*), yaitu perangkat lunak yang mempunyai tujuan sah tetapi berisi bug yang berbahaya. Hasil penelitian awal dari Symantec yang diterbitkan pada tahun 2008 menyatakan bahwa "kelajuan peluncuran kode yang berbahaya dan perangkat lunak lainnya yang tidak diinginkan, mungkin akan melebihi aplikasi perangkat lunak yang sah". Menurut F-Secure, "Jumlah perangkat perusak yang dibuat pada tahun 2007 sama dengan pembuatan dalam 20 tahun sekaligus". Jalur pembobolan perangkat perusak yang paling umum digunakan oleh penjahat kepada pengguna adalah melalui *Internet*, Surel dan *World Wide Web*.

**1.3. Sistem Android**

Android adalah sistem operasi yang berbasis linux untuk telepon seluler seperti smartphone dan tablet PC. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan oleh bermacam perangkat seluler [4]. Awalnya, *Google Inc* membeli *Android Inc.* pendatang baru yang membuat perangkat lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile* dan *Nvidia*.

Pada saat peluncuran pertama Android (5 November 2007), Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar *open source* pada perangkat seluler. Di pihak lain, Google merilis kode - kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open source* perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services (GMS)* dan kedua adalah yang benar - benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution (OHD)*.

Arsitektur grafis yang dimiliki oleh sistem android atau yang disebut "*Architecture of Android System*" seperti pada gambar 1, [1], [5].




**Gambar 1.** *Arsitektur Sistem Android*

***Applications***

Lapisan atas dari arsitektur android yang berisi aplikasi yang dikembangkan oleh pengembang android. Ada beberapa aplikasi standar yang, seperti *Browser* atau *SMS client*, namun pengguna dapat membeli dan menginstal aplikasi baru ke *Application Layer*.

***Application Framework***

*Application Framework* adalah lapisan kedua dalam arsitektur android. Aplikasi berkomunikasi langsung dengan *Application Framework*, yang cukup banyak menyediakan *tools* yang dibutuhkan untuk melakukan tujuan apa pun yang dirancang. Pengembang aplikasi langsung mengakses *Application Framework* untuk membangun fungsi dari aplikasi yang mereka buat. Selain aplikasi yang sebenarnya pada perangkat, *Application Framework* juga berkomunikasi dengan lapisan *Libraries* arsitektur android.

***Libraries***

*Libraries* asli android pada dasarnya hanya terdiri dari sejumlah fungsi yang memungkinkan perangkat untuk memproses berbagai jenis data. Beberapa *Libraries* ini khusus untuk jenis perangkat tertentu, serta dianggap *generic* untuk semua perangkat android.

***Android Runtime***

*Android Runtime* terdiri dari dua bagian besar, yaitu: *Core Libraries* dan *Dalvik Virtual Machine*. *Core Libraries* memungkinkan pengembang aplikasi android untuk membuat dan menyebarkan kode dalam bahasa pemrograman java. *Libraries Core* ini akan memiliki akses ke *Libraries* asli android serta *Dalvik Virtual Machine*. *Dalvik Virtual Machine,* fungsi aplikasi ini seolah-olah mesin mandiri dan mengeksekusi kode yang dibuat dengan *Java Core Libraries*. Hal ini juga berfungsi sebagai perantara antara *Java Core Libraries* dan *Hardware Abstraction Layer* dari perangkat Android

***Hardware Abstraction Layer***

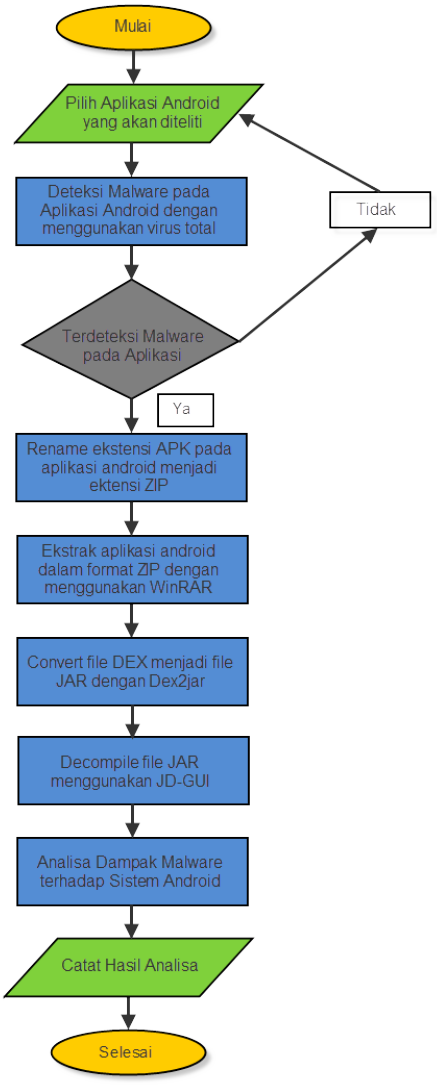
Beberapa diagram arsitektur android memiliki HAL yang termasuk bagian dari Linux Kernel. HAL pada dasarnya menangani komunikasi antara perangkat keras yang ditampilkan pada Linux Kernel dan semua lapisan perangkat lunak lain.

***Linux Kernel***

Sistem operasi Android pada dasarnya dibangun di atas Linux kernel 2.6 dan menyediakan *driver* yang dibutuhkan perangkat Linux untuk berkomunikasi dengan modul dari *Hardware Abstraction Layer*. Kernel Linux juga menangani semua fungsi sistem operasi dasar untuk perangkat android, seperti alokasi memori, komunikasi jaringan, dan keamanan aplikasi.

## 2. Pembahasan

### 2.1. Metode analisa *malware* android




**Gambar 2.** *Flowchart Metode Analisa Malware Android*

Dari Gambar 2 bisa dilihat alur metode yang digunakan dalam menganalisa *malware* android. Hal pertama yang dilakukan adalah memilih ataupun mencari aplikasi android yang akan diteliti, dalam hal ini aplikasi android di-download dari forum *contangio mobile*. Mendeteksi *malware* pada aplikasi android dengan menggunakan layanan virus total. Aplikasi android yang terdeteksi mengandung *malware* android akan dianalisa dengan cara *static analysis*. Aplikasi android yang terjangkit *malware* android berekstensi APK di-rename menjadi ekstensi ZIP, yang kemudian diekstrak dengan menggunakan WinRAR. Hal ini dilakukan karena file APK dapat di ilustrasikan sebagai sebuah *archive* (file ZIP) yang mengandung *Dalvix Executable* file (berekstensi DEX). Hasil dari ekstraksi berisikan beberapa file, dimana terdapat file berektensi DEX. Kemudian, file DEX di-convert kedalam format JAR dengan menggunakan Dex2jar dan akan menghasilkan file berekstensi JAR.

Langkah terakhir, file JAR di *decompile* menggunakan JD-GUI sehingga dapat dilihat semua *source code java* yang akan diteliti dan dianalisa untuk mencapai tujuan penelitian ini yaitu mendeteksi keberadaan *malware* android yang biasanya disisipkan pada *class file* aplikasi utama atau juga ada yang membuat root tersendiri, menganalisa cara kerja *malware* android ke dalam sistem android dan juga menganalisa dampak yang diberikannya terhadap sistem android.

Hasil penelitian ini diperoleh dari meneliti 3 sample aplikasi android yang terjangkit *malware* android, yaitu : *iCalendar, Live prints live wallpaper* dan *Hippo (kuis.com)*.

Pada penelitian ini, malware pada aplikasi android dideteksi dengan memanfaatkan layanan virus total, dimana secara umum analisa dan deteksi menggunakan beberapa metode, yaitu :

**a. *Signature Based:*** *Signature based* pada dasarnya bekerja pada pola biner. Nilai *hash* dari *malware* diidentifikasi dan disimpan pada database produk antivirus. Ketika mengeksekusi program baru dalam jaringan dan sistem membandingkan nilai *hash malware* yang tersimpan di database antivirus dan identifikasi apakah ini *malware* atau bukan. Masalah timbul ketika ada *malware* versi baru karena didalam penyimpanan hanya terdapat nilai *hash malware* versi lama. Untuk mencegah masalah ini, digunakan *generic signature*. Database produk generik menyimpan semua *signature malware* baru dan mengidentifikasikannya sebagai keluarga *malware*.

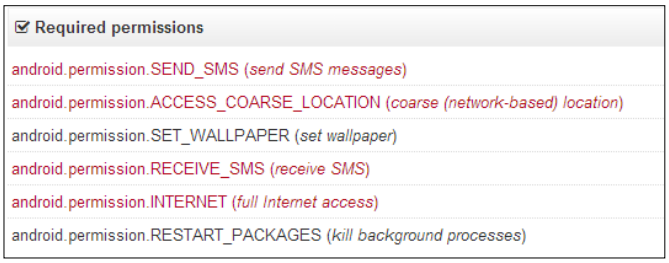
***b. Behavioral Based:*** *Behavioral based* bekerja pada lingkungan *virtual sandbox*. *Malware* didownload pada lingkungan sandbox ini dan bisa membuangnya tanpa membahayakan sistem dan data. *Malware* yang menyerang akan mengubah *signature* ketika antivirus pelindung mendeteksinya. Pada teknik ini *malware* dideteksi menggunakan dua cara, pertama melewati antivirus berbasis host dan kedua melewati gateway antivirus.

**c. *Anomaly Based****; anomaly based* mendeteksi perilaku user. Apabila terdapat perubahan perilaku user dalam jaringan maka dilakukan perbandingan *signature* yang sebelumnya disimpan dengan signature di *database antivirus*. Pendekatan deteksi *anomaly based* digunakan dua tahap. Tahap pertama adalah tahap latihan yang mengidentifikasi perilaku sistem terhadap kehadiran penyerang dan teknik pembelajaran mesin. Tahap kedua adalah membandingkan perilaku user terhadap perilaku user saat ini. Apabila terdapat beberapa perubahan terhadap perilaku user saat ini maka identifikasi apakah

### 2.2. Analsis *malware iCalendar*

Didalam aplikasi iCalendar yang berfungsi sebagai kalender elektronik pada perangkat android dengan beberapa tambahan fitur juga disisipkan suatu *malware* yang diduga dapat mengirimkan SMS ke premium number sehingga pulsa dari perangkat android yang terjangkit berkurang tanpa diketahui oleh user.

Sebelumnya, dengan bantuan layanan Virus Total peneliti akan mendeteksi apakah aplikasi iCalendar terjangkit malware android. Dari hasil analisa menggunakan layanan virus total terdeteksi bahwa terdapat *malware* pada aplikasi iCalendar.apk dan juga terdapat kejanggalan pada beberapa *required permissions* (Gambar 3).



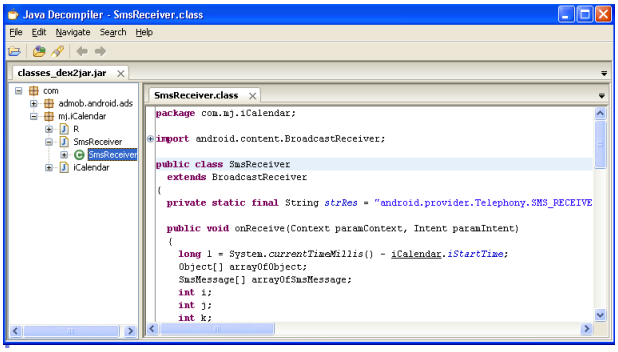

**Gambar 3.** *Detail permission file iCalendar.apk*

Peneliti melakukan analisa terhadap aplikasi iCalendar.apk dengan menggunakan beberapa *tools* komputer forensik. Hal pertama yang dilakukan adalah rename ekstensi APK menjadi ekstensi ZIP. Kemudian file iCalendar.zip diekstrak menggunakan *software WinRAR* dan akan menghasilkan beberapa file serta terdapat juga file berekstensi DEX.

Menggunakan software dex2jar, file DEX di-*convert* menjadi file JAR dengan perintah di jendela *command prompt* dan menghasilkan file JAR

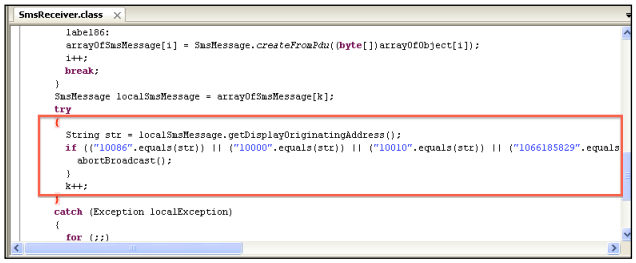
Langkah terakhir adalah decompile file JAR dengan menggunakan JD-GUI sehingga dapat terlihat keseluruhan *source code java* dari aplikasi iCalendar.apk. Terlihat pada gambar 4, terdapat *class file*

dengan nama *SmsReceiver.class* yang seharusnya tidak dibutuhkan dalam aplikasi iCalendar.




**Gambar 4.** *class file SmsReceiver.class*

Setelah dianalisis *source code* pada SmsReceiver.class. terdapat *Local SMS* dengan *originating address* dan terdapat beberapa nomor yang mencurigakan *(Gambar 5)*. Selain itu, terdapat perintah *abort broadcast* yang ditujukan sebagai cara agar menolak laporan penerimaan SMS ke perangkat android, sehingga user tidak mengetahui adanya aktifitas SMS yang berjalan.




**Gambar 5.** *Analisa SmsReceiver.class*

Untuk lebih jelas, peneliti juga menganalisa *source code* yang ada pada *class file iCalendar.class*. Pada *Gambar 6*, terlihat lagi perintah send SMS. Setelah peneliti menelusuri *source code-nya*, ditemukan lagi satu nomor dan ini sama dengan nomor yang peneliti temukan pada *source code SmsReceiver.class*.


**Gambar 6.** *Analisa iCalendar.class*

Setelah diteliti lagi dengan menggunakan layanan *search engine Google*, ternyata nomor "1066185829" merupakan mobile service provider (SP) number dari operator *China Mobile* (Gambar 7).


**Gambar 7.** *mobile service provider number China*

Dari beberapa analisa yang dilakukan peneliti terhadap *source code java* aplikasi iCalendar.apk, peneliti merangkum cara kerja *malware* yang menjangkit aplikasi iCalendar.apk. *Malware* tersebut aktif setelah aplikasi iCalendar.apk terpasang pada perangkat android. *Malware* ini mengirimkan sekali *sms* ke *premium number* 1066185829 dengan isi SMS 921X1. Oleh sebab itu, SMS dari *premium number* masuk ke perangkat android yang terjangkit *malware* ini.

Pada penjelasan di atas tadi juga telah dijelaskan bahwa adanya penolakan laporan pengiriman dan penerimaan SMS dari premium number 1066185829 ke perangkat android, sehingga user sama sekali tidak mengetahui adanya aktifitas SMS yang akan berdampak pada berkurangnya pulsa yang dimiliki user.


**Gambar 8.** *Info aplikasi iCalendar pada perangkat android*

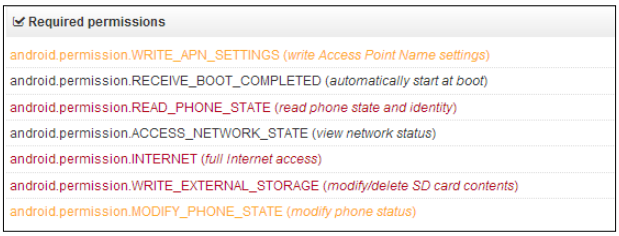
### 2.3. Analsis *malware Live prints live wallpaper*

*Live prints live wallpaper* merupakan aplikasi *live wallpaper* untuk perangkat android dengan tampilan *live prints*. Model *live prints* ini sangat bagus untuk dijadikan *live wallpaper* pada perangkat android, tetapi ternyata

aplikasi ini juga mengandung *malware android* yang disebut *Hongtoutou (Trojan.spy.adrd A)*.


**Gambar 9.** *Aplikasi liveprintslivewallpaper.apk*

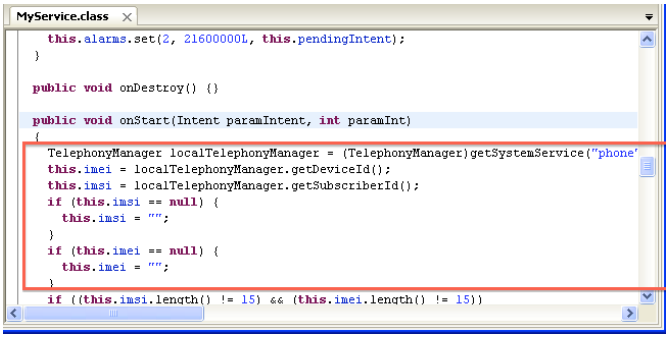
Jika dilakukan analisa dengan menggunakan layanan virus total, ditemukanlah *malware android* yang menyusup ke dalam aplikasi *live prints live wallpaper*. *Malware android* yang terdeteksi adalah *Hongtoutou* (*Trojan spy adrd A*) yang dapat membaca dan mencuri status atau identitas pribadi pada perangkat android




**Gambar 10.** *Detail file liveprintslivewallpaper.apk*

Setelah ekstensi APK dirubah menjadi ekstensi ZIP. Kemudian, *liveprintslivewallpaper.zip* diekstrak dengan menggunakan software WinRAR. Hasil ekstrak terdiri beberapa yang file dan termasuk file berekstensi DEX pada salah satu filenya. Untuk melakukan analisa *source code java* dari aplikasi android, file berektensi DEX di-*convert* menjadi file JAR dengan memanfaatkan *tool* Dex2jar .

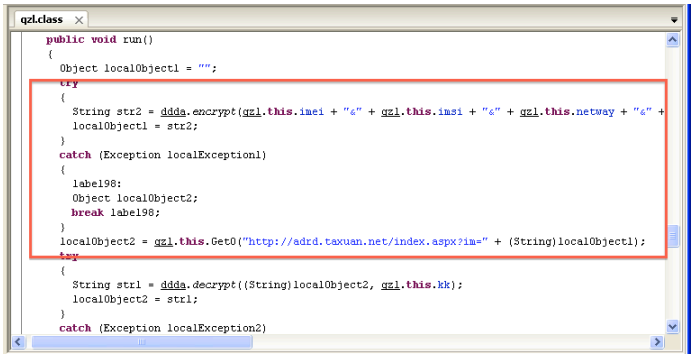
Setelah didapat file berekstensi JAR kemudian dilakukan *decompile* menggunakan *tool JD-GUI* untuk menampilkan secara keseluruhan *source code java* dari aplikasi liveprintslivewallpaper.apk.




**Gambar 11.** *Perintah untuk melihat IMEI dan IMSI*

Pada Gambar 11, terlihat *source code* yang berisikan "*TelephonyManager*" yang dimaksudkan untuk melihat *IMEI (International Mobile Equipment Identity)* dan *IMSI (International Mobile Subscrilber Identity)* perangkat android yang terjangkit *malware android hongtoutou*.

Setelah *malware android Hongtoutou* ini membaca IMEI dan IMSI, selanjutnya hongtoutou mengambil informasi tentang IMEI dan IMSI tersebut. IMEI dan IMSI merupakan identitas pribadi yang dimiliki setiap perangkat seluler termasuk perangkat android serta IMEI dan IMSI ini berbeda untuk setiap perangkat seluler. Informasi tentang IMEI dan IMSI pada perangkat android dikirimkan ke sebuah situs tak dikenal yaitu http://adrd.taxuan.net/index.aspx (Gambar 12).




**Gambar 12.** *Script Mencuri IMEI dan IMSI perangkat android*

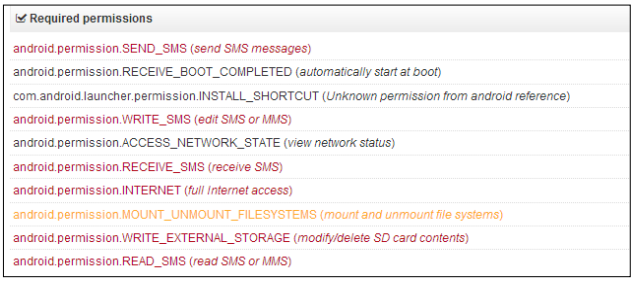
Pada kasus aplikasi *live prints live wallpaper*, *malware android* mulai bekerja saat aplikasi *liveprintslivewallpaper.apk* dipasangkan dan dijalankan pada perangkat android. *Malware android Hongtoutou* pun mulai membaca dan mengambil informasi IMEI dan IMSI perangkat android yang dijangkitinya tanpa diketahui user perangkat android tersebut.


**Gambar 13.** *Info aplikasi live prints live wallpaper pada perangkat android*

### 2.4. Analsis *malware Hippo (kuis.com)*

Didalam aplikasi Hippo yang merupakan aplikasi android yang terhubung dengan situs kuis.com. Aplikasi ini normalnya memberikan layanan kuis yang dilengkapi juga media *chat, search engine* dan yang lainnya. Akan tetapi, aplikasi android ini diduga mengandung *malware* android yang disebut ANDROIDOS_HIPPOSMS.A dan lebih terkenal dengan nama Hippo SMS.

Untuk memastikan adanya *malware* android pada aplikasi Hippo SMS, peneliti melakukan analisa awal dengan memanfaatkan layanan Virus total. Pada Gambar, terdapat 35 dari 42 antivirus yang mendeteksi adanya *malware* android pada aplikasi Hippo SMS. Selain itu, dapat dilihat juga ada beberapa *permission* yang dianggap tidak seharusnya diperlukan pada aplikasi android Hippo_sample.apk ini. (Gambar 14)



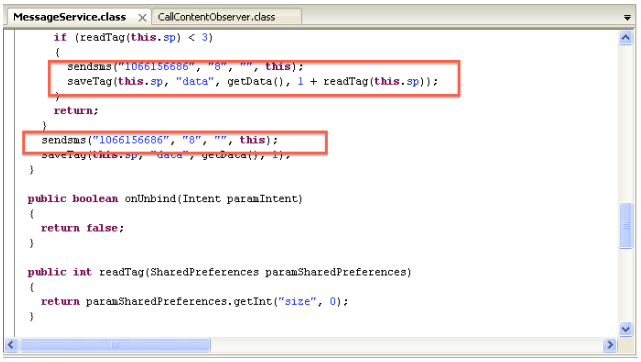

**Gambar 14.** *Detail file Hippo_sample.apk*

Seperti proses penelitian sebelumnya, peneliti merubah ekstensi *Hippo_sample.apk* menjadi *Hippo_sample.zip* setelah itu diekstrak dengan menggunakan *tools* WinRAR dan memperoleh file *classes.dex*.

Langkah selanjutnya adalah *convert* format DEX pada file *classes.dex* menjadi format JAR dengan menggunakan *tools Dex2jar*. Dengan operasi *convert* menggunakan Dex2jar pada *command prompt*, didapatlah file dengan format JAR.

Untuk menganalisa *malware android hippo*, bisa dilakukan dengan menganalisis *source code java* lengkap dari aplikasi android yang terjangkit *malware* android dengan memanfaatkan *tools JD-GUI*. Terlihat terdapat paket *class file* dengan nama "sms", yang didalamnya terdapat *3 class file* yang berisikan *malicious code*, yaitu *BootReceiver.class, CallContentObserver.class, MessageService.class*.

Pada class file *MessageService.class* terdapat perintah *"sendsms"* yang ditujukan pada suatu nomor yaitu "1066156686" dengan text sms "8" (Gambar 15). Dengan menggunakan layanan *premium number checker* pada situs www.sohao.org, terlihat bahwa nomor "1066156686" merupakan *mobile service provider* dari *operator china mobile* (Gambar 16). Sehingga dapat diperjelas lagi bahwa sms dengan text "8" ke nomor "1066156686" yang dilakukan malware android menyebabkan perangkat android berlangganan sms premium.

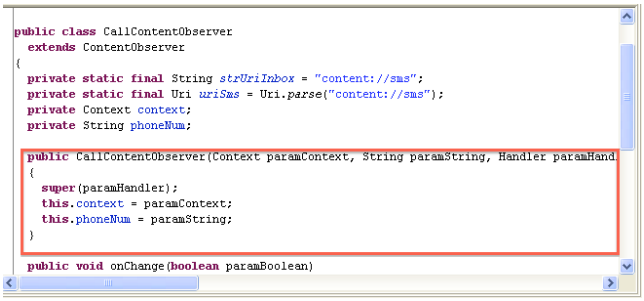



**Gambar 15.** *Analisa class file MessageService.class*


**Gambar 16.** *Pengecekan premium number di www.sohao.org*

Pada *class* file *CallContentObserver.class* terdapat variabel *"PhoneNum"*, dimana data nomor telepon dan juga *premium number* termasuk ke dalam variabel *"PhoneNum"* (Gambar 17). Peneliti kembali melihat class file *MessageService.class*, terdapat *"notificationManager"*. *"notificationManager"*

dihubungkan dengan *class* file *CallContentObserver.class*, yang mana dapat peneliti jelaskan bahwa "*PhoneNum*" yang diawali dengan angka "10" maka notifikasi sms masuk ataupun keluar ditolak sehingga user perangkat android tidak mengetahui adanya aktifitas sms yang terjadi pada perangkat androidnya (Gambar 18).

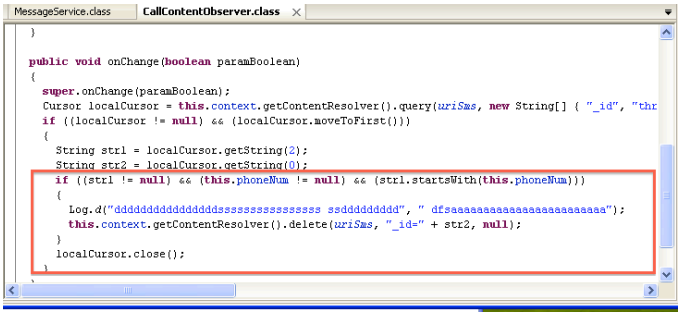



**Gambar 17.** *Analisa class file CallContentObserver.class*


**Gambar 18.** *Terdapat penolakan notifikasi dari PhoneNum yang diawali angka "10*

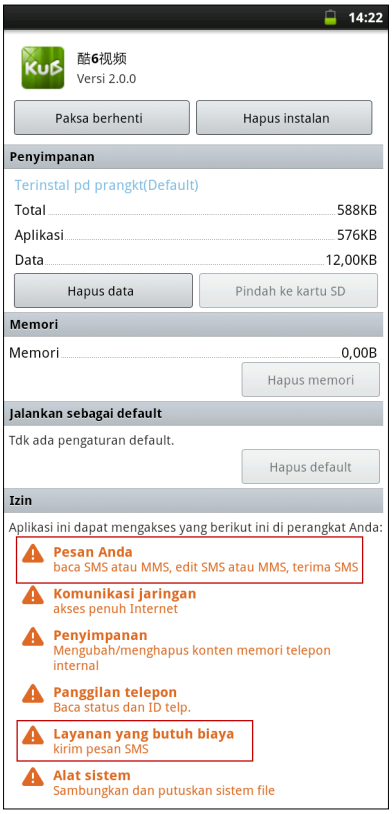
Selain itu, untuk menyempurnakan aktifitas *malware android Hippo SMS* agar sama sekali tidak disadari oleh user, *malware android* ini juga memberikan perintah untuk menghapus setiap sms masuk dari *china premium number*. Pada class file *CallContentObserver.class* dijelaskan apabila terdapat *sms* masuk dari "PhoneNum" dimana yang diawali dengan angka "10" (seperti yang dijelaskan di atas), maka dengan otomatis *malware* android ini memberikan perintah *reply sms* ke premium number dengan text "ddddddddddddddddsssssssssssssssssddddddddd" sehingga sms masuk yang asli dari premium number terhapus (Gambar 19). Hal ini menyebabkan user perangkat android sama sekali tidak mengetahui adanya aktifitas sms premium yang berakibat berkurangnya pulsa yang dimiliki user.

Pada aplikasi Hippo ini, *malware* android akan aktif setelah aplikasi Hippo dipasangkan ke perangkat android dan tentunya saat perangkat android terkoneksi dengan *provider* telekomunikasi. Saat *malware android* aktif, aktifitas pengiriman sms ke *premium number* "1066156686" dengan text "8" langsung dijalankan dan mengakibatkan perangkat android yang terjangkit *malware android* ini berlangganan sms premium dari berbagai *chinese premium number*. Pada Gambar 4.40, dapat dilihat aplikasi Hippo yang sudah terpasang pada perangkat android.




**Gambar 19.** *Perintah untuk menghapus sms dari premium number*

Pada Gambar 20, terlihat info aplikasi yang berisikan beberapa permission yang tidak seharusnya diperlukan oleh aplikasi Hippo_sample.apk.




**Gambar 20.** *Info aplikasi Hippo (kuis.com) pada perangkat android*

## 3. Kesimpulan

Dari hasil analisis yang dilakukan dapat diambil beberapa kesimpulan, yaitu:

1. Proses analisis *malware* forensik pada aplikasi android bisa dilakukan dengan mengekstrak file apk aplikasi, kemudian menkonvertnya dengan *tools dex2jar* dan JD-GUI sehingga hasil akhir didapatkan *source code java* dari aplikasi yang bisa dianalisis.
2. Aplikasi iCalender mengandung *malware* android (*trojan*) yang dapat mengirimkan sms ke premium

number “1066185829”, sehingga tanpa disadari user akan menerima sms premium dari premium number tersebut.

3. Aplikasi *live prints live wallpaper* mengandung *malware android hongtoutou* (*Trojan* dan *spyware*) yang dapat mencuri informasi tentang IMEI dan IMSI dari perangkat android dikirimkan ke situs http://adrd.taxuan.net/index.aspx .
4. *Malware* android Hippo sms yang terdapat dalam aplikasi hippo membuat perangkat android mengirim sms ke salah satu *premium number* China (“1066156686”) dan mengakibatkan perangkat android tersebut berlangganan sms premium dari berbagai *premium number* China tanpa diketahui user.
5. Bagi para pengguna android sebaiknya mendownload aplikasi android dari situs terpercaya, dan sebelum memutskan menginstal aplikasi sebaiknya memeriksa terlebih dahulu dengan anti virus apakah aplikasi tersebut mengandung malware, serta terlebih dahulu memeriksa izin akses yang dijalankan oleh aplikasi.

## Daftar Pustaka

[1] A. Hoog, "Android Forensics: Investigation, Analysis and Mobile Security for Google Android ". Syngress, 2011.

[2] Afrianto, D. Setyo, “Antisipasi Cybercrime Menggunakan Teknik Komputer Forensik”, *Jurnal Informatika Teknologi Industri, Universitas Islam Indonesia. 2007*.

[3] G.Kaur, B. Nagpal, “Malware Analysis & its Application to Digital Forensic”, in *International Journal on Computer Science and Engineering (IJCSE),* pp. 622-626*,* Vol. 4 No. 04 April 2012.

[4] K. Sharma,T. Dand, T. Oh, W. Stackpole, “Malware Analysis for Android Operating”, *8th Annual Symposium on Information assurance* , pp. 31-35, June 4-5, 2013

[5] V. Manjunath, “Reverse Engineering Of Malware On Android,” *in SANS Institute InfoSec Reading Room., University of Essex*, August 2011.

[6] V. Svajcer*, “*Sophos Mobile Security Threat Report*,” Launched at Mobile World Congress*, February 2014.

[7] W.G. Kruse, J.G. Heiser, “Computer Forensics Essentials”, Adisson Wesley, 2002.