ANALISIS FORENSIK *MALWARE POP-UP ADS* IKLAN PADA PLATFORM *ANDROID*

Harsono[1], Vivi Sahfitri, S.Kom., M.M[2], Ferdiansyah, M.Kom[3]

Mahasiswa Universitas Bina Darma[1], Dosen Universitas Bina Darma[23]

Jl. Jenderal Ahmad Yani No.03 Plaju, Palembang 30264

harsonobae123@gmail.com[1], vivi_sahfitri@binadarma.ac.id[2], ferdi@binadarma.ac.id[3]

Abstrack : The use of smartphones with Android platform as an open source operating system is very widespread. However, because Android is an open source system, it is easier for anyone to develop Android applications that can be downloaded on Google Play Store and others. Malware developers usually insert a guard in an application that is ready for users to download, sometimes they also insert in pop-up ads which when clicked will open a new window. Analyze the android application using forensic computer techniques. This analysis phase starts with renaming the APK to zip, then extending the zip file so that it becomes the DEX format. The DEX file is converted to a JAR file with the Dex2jar tool. Then the results of the JAR file are decompiled with JD-GUI to see the java source code which is then analyzed. The results of the analysis show that Android malware works by sending an SMS without the user's knowledge and also stealing information in the form of IMEI and IMSI belonging to the user.

Keyword : Malware, Android, Pop-up Ads, Computer forensics

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 google play store dan lain-lain. Para pengembang malware biasanya menyisipkan mawalre didalam aplikasi yang siap diunduh oleh user, terkadang mereka juga menyisipkan di pop-up ads yang mana ketika diklik akan membuka jendela baru. Analisa terhadap aplikasi android tersebut dengan menggunakan teknik komputer forensik. Tahapan analisis ini dimulai dengan merename apk menjadi zip, kemudian mengekstark file zip 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. Hasil analisis memperlihatkan malware android bekerja dengan mengirim sms tanpa sepengetahuan user dan juga mencuri informasi berupa IMEI dan IMSI milik user.

Kata kunci : *Malware, Android Pop-up Ads, Komputer Forensik*

1. PENDAHLUAN

Kemajuan teknologi saat ini dengan berbagai jenis *gadget* seperti *tablet PC* dan *smartphone*, tentunya didukung oleh beberapa jenis sistem operasi. Sejalan dengan hal tersebut, salah satu sistem operasi yang sedang marak digunakan untuk mendukung kerja *tablet PC* ataupun *smartphone* adalah sistem android. *Android* menurut Safaat (2015:2) merupakan sistem operasi untuk telpon seluler yang berbasis *linux*. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

Malware atau juga disebut *software* perusak mencakup *virus* komputer, *trojan horse*, perangkat pengintai (*spyware)*, perangkat iklan (*adware*), dan perangkat jahat (*crimeware*). *Pop-up* jenis iklan yang akan muncul dalam jendela baru diatas jendela halaman *web* yang sedang

kita lihat. Terkadang *pop-up* juga bisa menimbulkan *malware* ketika kita mengklik iklan yang muncul. Ada beberapa kasus penyerangan *malware pop-up* dalam waktu dekat ini, korbannya adalah para pengguna *android*. Dikutip dari *web* resmi milik perusahan keamanan *cyber checkpoint* pada tanggal 25 mei 2017 mereka berhasil menemukan keberadaan *malware* yang menginfeksi perangkat *android* yaitu *malware Judy*.

Disebut sebagai *Judy*, *malware* ini tertanam dibeberapa aplikasi yang tersedia di *Google Play Store*. Isinya adalah *adware auto clicking*, *Judy* menghasilkan klik palsu pada sebuah iklan. Hasli penelitian Checkpoint menunjukan sekitar 36,5 juta perangkat *android* di seluruh dunia sudah terinfeksi *malware* ini. Pada tahun berikutnya tepatnya pada tanggal 12 januari 2018 para peneliti perusahaan keamanan *cyber* checkpoint kembali menemukan *malware* jenis baru, yaitu *malware AdultSwine*. *Malware* ini tersembunyi didalam 60 aplikasi *game* yang di peruntukkan untuk anak-anak. *Malware* ini akan menyusup atau disusupi ke dalam aplikasi dan bekerja menampilkan iklan *pop-up* yang memenuhi seluruh layar ponsel, lalu mencoba memberi ajakan agar pengguna mengunduh aplikasi *antivirus* palsu atau mendaftar layanan SMS premium.

2. LANDASAN TEORI

2.1. *Malware*

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. Dengan semakin pesatnya perkembangan teknologi yang terus memicu dikembangkannya *malware* baru, sehingga semakin banyak pihak yang dirugikan karena adanya *malware-malware* ini, saat ini *malware* dapat menjangkiti hampir seluruh jenis sistem operasi termasuk sistem operasi *android*. (Ferdiansyah, 2018).

2.2. Komputer Forensik

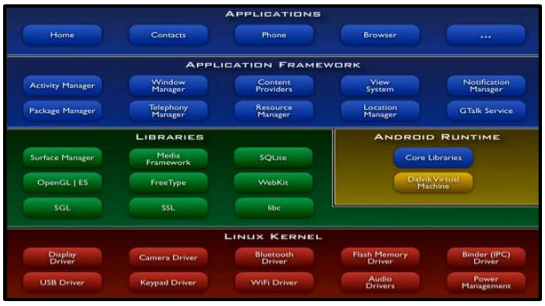
Menurut Feri Sulianta (2012: 4) Komputer forensik merupakan suatu rangkaian metodelogi yang terdiri dari teknik dan prosedur untuk mengumpulkan bukti-bukti berbasis entitas maupun piranti digital agar dapat dipergunakan secara sah sebagai alat bukti di pengadilan. Komputer forensik menjadi bidang ilmu baru yang mengawinkan dua bidang keilmuan, yakni ilmu hukum dan komputer. Komputer forensik atau digital forensik banyak ditempatkan dalam berbagai keperluan, bukan hanya kasus-kasus kriminal yang melibatkan hukum, bahkan berguna untuk kebutuhan khusus lain sehubungan pekerjaan yang melibatkan teknologi informasi.

2.3. *Android*

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux Yang mencakup *sistem operasi*, *middleware* dan *aplikasi*. *Android* menedikan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google inc.* membeli *Android inc*. yang merupakan pendatang baru yang membuat peranti lunak untuk

ponsel/*smartphone.* Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.*

Pencetus gagasan lahirnya *android* dimulai oleh *Google Inc* yang berkolaborasi dengan *Android Inc* yang merupakan perusahaan pendatang baru pembuat peranti perangkat lunak untuk ponsel. Sedangkan untuk mengembangkan dan memperoleh sistem operasi *android*, maka dibentuklah *Open Handset Alliance*. Pada saat perilisan perdana *android,* sekitar tahun 2007, *android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler, *Google* merilis kode-kode *android* dibawah lisensi *apache*. Sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. (Safaat, 2015:1).



Gambar 2.1 Arsitektur *Android*

Pada gambar 2.1 (Damarullah, 2013) merupakan susunan dari arsitektur pada *android* yang terdiri dari *Linux Kernel*. *Kernel linux* menyediakan driver layar, kamera, *keypad*, *WiFi, Flash Memory*, *audio*, dan *IPC* (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan. *Library* yang terdiri dari Media *Library* dan *Graphic Library*, kemudian *Android Runtime* yang merupakan mesin *virtual* yang membuat aplikasi *android* menjadi lebih tangguh dengan paket pustaka yang telah ada.

Dalam *Android Runtime* terdapat 2 bagian utama, yakni pustaka inti dan mesin *virtual dalvik*. *Application Framework* yaitu kerangka aplikasi yang menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi *android*. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan user interface dan sumber daya aplikasi. *Applications* yang merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam *Android runtime* dengan menggunakan kelas dan *service* yang tersedia pada *framework* aplikasi (Damarullah, 2013).

2.4. *Pop-Up Ads*

Pop-up ads merupakan jenis iklan yang akan muncul dalam jendela halaman *website* yang sedang dikinjungi. *Pop-up ads* biasanya juga muncul pada video yang sedang di tonton. *Pop-up ads* sering dibilang spam/sampah ketika sedang mengunjungi sebuah *website.* Tak heran *pop-up ads* juga bisa menimbulkan hal yang dapat merugikan misalnya *malware*, *multiple pop-ups* dan konten dewasa. (Oktaviani, dkk. 2017).

3. METODELOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan oleh penulis kurang lebih selama 6 bulan, mulai dari bulan Maret 2018 sampai dengan bulan Agustus 2018 dengan mengumpulkan data yang tepat sesuai dengan penelitian.

3.2 Data Penelitian

Adapun data-data penelitian yang dikumpulkan untuk proses analisis *malware* pada platform *android* ini diambil dari beberapa sampel aplikasi yang nantinya akan di lakukan analisis.

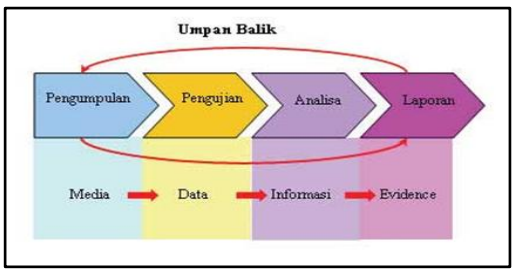
3.3 Metode Penelitian

Pada penelitian ini apalikasi pada *android* dideteksi dengan memanfaatkan layanan *virus* total, dimana secara umum pengujian, analisa dan deteksi menggunakan metode komputer forensik :

1. Pemodelan Forensik

   Model forensik melibatkan tiga komponen guna mencapai kelayakan dan hasil yang berkualitas yaitu meliputi manusia (*people)* sebagai penyidik, peralatan *(equipment),* dan aturan *(protocol)*.

   Ada empat fase yang dilakukan dalam komputer forensik, antara lain :



Gambar 3.1 Tahapan Komputer Forensik

2. Pengumpulan Data
   a. Pengumpulan data adalah langkah pertama dalam proses forensik untuk mengidentifikasi sumber – sumber potensial dan menjelaskan langkah pengumpulan data.
   b. Data bertumpu pada personal komputer, mobile komputer, jaringan komputer, media penyimpanan dan integrasi penyimpanan.
3. Pengujian
   a. Melakukan pengujian, menilai dan mengekstrak kepingan informasi yang relevan dari data – data yang dikumpulkan.
   b. Tahapan ini melibatkan bypassing fitur –fitur sistem, filtrasi (eliminasi data), meng-exclude file, mengalokasi file dan mengekstrak file.
4. Analisa

Begitu informasi diekstrak, examiner melakukan analisis untuk merumuskan kesimpulan dalam menggambarkan data. Analisis yang dimaksud mengambil pendekatan metodis untuk menghasilkan kesimpulan yang berkualitas didasarkan pada ketersediaan data atau bahkan sebaliknya.

Menurut Heriyanto (2016:284) ada 5 teknik analisa sebagai berikut :

a. *Signature-based malware detection*

Terdapat pola *(pattern)* berupa *'squence of bytes'* dalam kode program yang umum terdapat dalam satu *malware* atau *hash value* dari *known bad files* yang dijadikan acuan bagi pengembang *antivirus* untuk mendeteksi keberadaan *mobile malware*.

b. *Specification-based malware detection*

Pengembang *antivirus* akan menetapkan satu set aturan *(rules)* yang menjadi acuan bagi satu aktivitas normal dari aplikasi, sehingga jika terjadi aktivitas diluar parameter normal tersebut, maka satu aplikasi dikategorikan sebagai *malicious.*

c. *Behavioral-based detection*

Aplikasi *antivirus* ini tidak saja melakukan proses *scanning* di permukaan saja, tetapi juga mengecek aktivitas *malware.* Teknik ini akan menghasilkan *database* atas aktivitas/perilaku *malicious* dengan cara mempelajari sejumlah kelompok *(family) malware.* Umumnya, terdapat satu kelompok *malware* yang awalnya berasal dari satu *malware* yang mengalami perkembangan hingga terdapat banyak *malware* dengan 'perilaku' yang relatif sama. Deteksi berdasarkan perilaku ini mampu mendeteksi *metamorphic malware* yang dapat bereproduksi.

5. Laporan

a. Laporan adalah tahapan akhir dari proses komputer forensik. Dalam tahap ini kita mempresentasikan informasi yang merupakan hasil dari proses analisis.

b. Faktor yang mempengaruhi reporting adalah alternative consideration (penjelasan alternatif), audience consideration (pertimbangan pengamat) dan actionable information.

3.4 Metode Pengumpulan Data

Untuk mendukung kebutuhan data yang dibutuhkan selama proses penelitian berlangsung, penulis menggunakan beberapa metode dalam pengumpulan data, yaitu :

1. *Observasi,* merupakan metode pengumpulan data dimana peneliti mencatat informasi sebagaimana yang disaksikan selama penelitian.
2. Dokumentasi, merupakan suatu metode pengumpulan data tentang hal yang berkaitan dengan penelitian.
3. Studi Pustaka *(Literature),* merupakan metode pengumpulan data yang dilakukan oleh peneliti dengan menelaah teori-teori, pendapat serta pokok pikiran yang menunjang dengan masalah yang dibahas.

3.5 Daftar Aplikasi

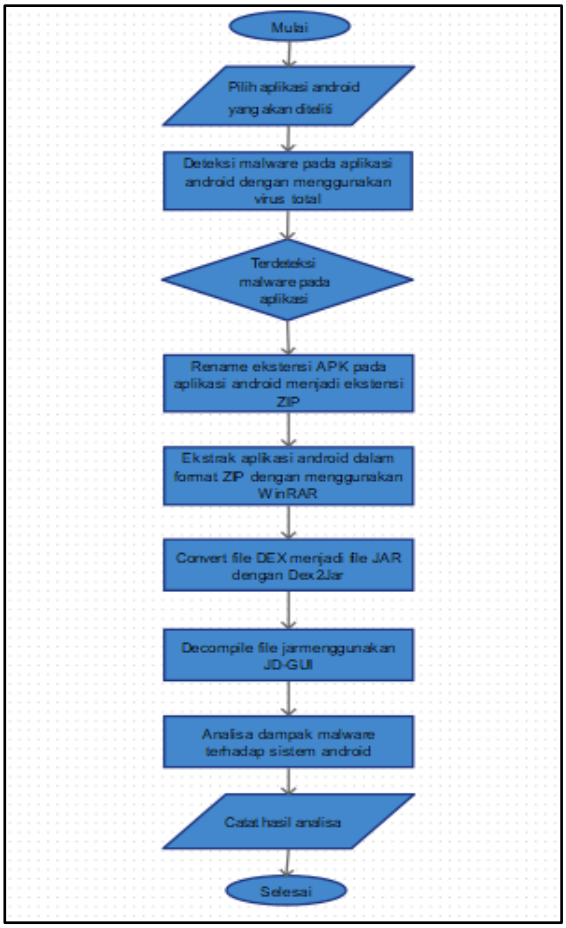
Tabel 3.1 Aplikasi yang dianalisa

No	Nama Aplikasi
1	ApkCustom
2	BadNews
3	Bigo Live
4	Bima +
5	Clan War
6	Chef Judy Picnic Lunch Maker

7	Euro Truck Simulator
8	Fastest Browser
9	GlamorousSmoke
10	Hentai Manga
11	Hippo (Kuis)
12	King Of Glory
13	Kuota Gratis
14	Launcher For Terraria Mods
15	Live Prints Live Wallpaper
16	Lords Scared Heavenly Cross
17	Mobidash
18	MySmartfren
19	Tokopedia
20	Wa Pink Kitty Cantik

Tabel 3.1 adalah aplikasi yang akan di lakukan analisa, sebelumnya telah di pilih 100 aplikasi dan hanya 20 yang akan di ambil yakni dari daftar tabel 3.1 diatas.

3.6 Metode Analisa *Malware*



Gambar 3.2 *Flowchart* metode analisis

Dari gambar 3.1 bisa dilihat alur yang digunakan dalam menganalisa *malware android.* Hal pertama yang dilakukan adalah memilih atau mencari aplikasi *android* yang akan diteliti. Langkah selanjutnya 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.*

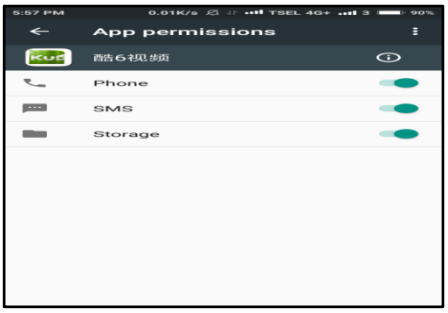
. Hal ini dilakukan karena file *APK* dapat diilustrasikan sebagai sebuah *archive* (file *ZIP*) yang mengandung *Dalvix Executable* file (berekstensi *DEX*). Hasil dari ekstraksi berisikan beberapa file, dimana terdapat file berekstensi *DEX*. Kemudian, file *DEX* di-*convert* kedalam format *JAR* dengan menggunakan *Dex2jar* dan akan menghasilkan file berekstensi *JAR*. Langkah terakhir yaitu, 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* kedalam sistem *android* dan juga menganalisa dampak yang diberikannya terhadap sistem *android.*

4. HASIL

Penelitian yang dilakukan oleh peneliti, dipaparkan dari tahapan awal penelitian, proses penelitian hingga hasil dari penelitian. Hasil dari penelitian ini di peroleh 2 sampel aplikasi *android* yang terindikasi *malware*. Peneliti telah melakukan penelitian 20 sampel aplikasi *android*, tetapi hanya 2 aplikasi yang terdapat *malware* pada aplikasi tersebut yaitu Hippo (Kuis) dan *Live Prints Live Wallpaper*.

4.1 Hippo (Kuis)

Aplikasi Hippo (Kuis) merupakan aplikasi *android* yang memberikan layanan kuis yang dilengkapi juga media *chat*, *search engine* dan lainnya. Akan tetapi, ketika di deteksi menggunakan layanan *virus* total terdapat adanya *malware Android.Trojan.HippoSMS.A.*

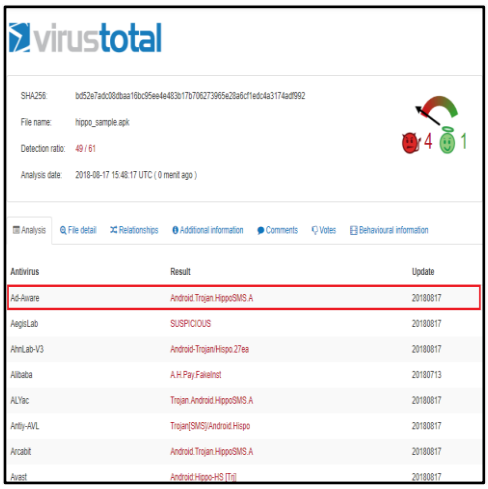


Gambar 4.1 Izin Aplikasi Hippo (Kuis)

Gambar 4.6 merupakan izin dari aplikasi hippo (kuis) dimana terdapat izin yang tidak diperlukan oleh aplikasi ini yaitu SMS.

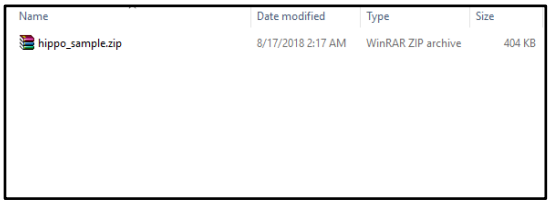
Gambar 4.2 Peringatan dari apliksi Hippo (Kuis)

Gambar 4.2 merupakan tampilan setelah di install dan terdapat peringatan bahawa aplikasi tersebut mengandung *virus*.



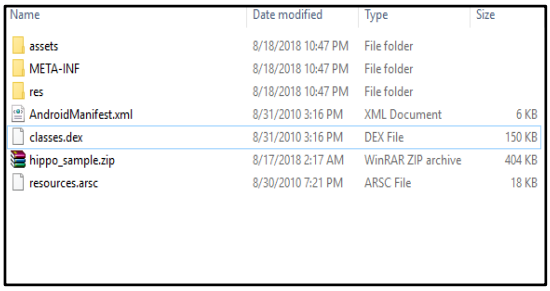
Gambar 4.3 Detail *Malware* Aplikasi Hippo

Selanjutnya adalah merubah ekstensi file Hippo.apk menjadi .zip seperti gambar 4.4 di bawah ini.



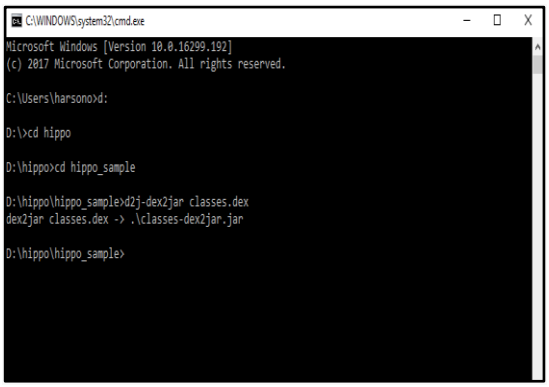
Gambar 4.4 *Rename* ekstensi Hippo (Kuis).zip

Kemudian ekstrak file Hippo (Kuis).zip menggunakan *WinRAR* dan akan menghasilkan beberapa file seperti (Gambar 4.5) serta terdapat juga file yang berekstensi DEX.

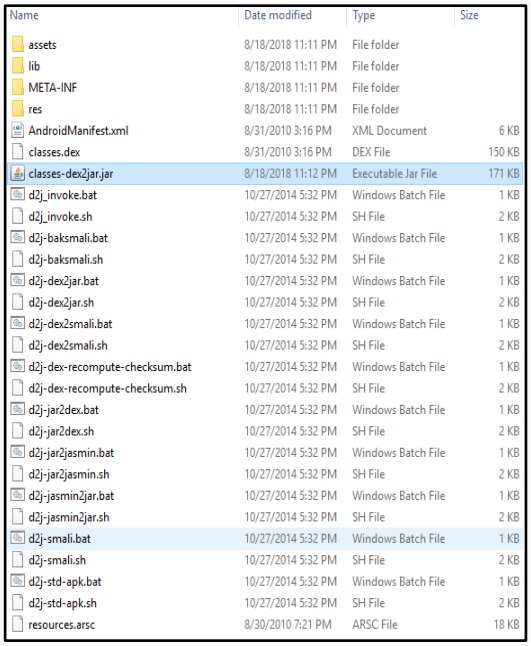


Gambar 4.5 Hasil ekstrak file Hippo (Kuis).zip

Selanjutnya menggunakan *software* dex2jar, file yang berekstensi .DEX di-*convert* menjadi file JAR dengan memanfaatkan jendela *command prompt* (Gambar 4.6) dan menghasilkan file JAR (Gambar 4.7).

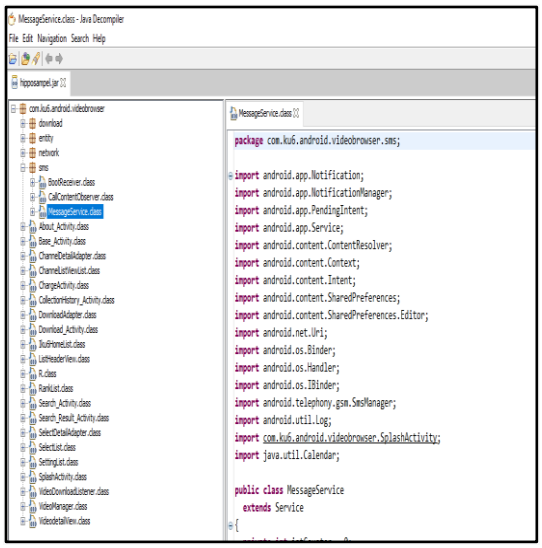


Gambar 4.6 Operasi *convert* dex2jar



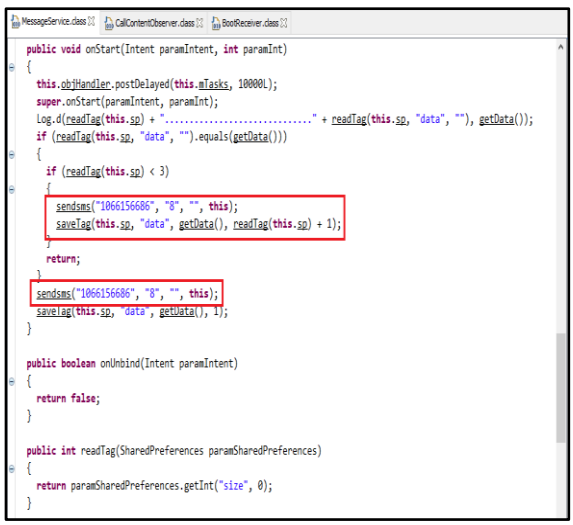
Gambar 4.7 Hasil *convert* menggunakan dex2jar

Langkah terakhir adalah *decompile* file JAR dengan menggunakan JD-GUI sehingga dapat terlihat keseluruhan *source code java* dari aplikasi Hippo (Kuis).apk. Dapat dilihat pada gambar 4.8 dibawah ini terdapat *class file* dengan nama *MessageService.class* yang seharusnya tidak dibutuhkan dalam aplikasi Hippo (Kuis).



Gambar 4.8 Terdapat *MessageService.class*

Oleh karena itu, peneliti melakukan analisa terhadap *source code* pada *MessageService.class.* Setelah di lihat dengan teliti terdapat perintah "*sendsms"* yang ditujukan pada suatu nomor yaitu "1066156686" dengan text sms "8" (Gambar 4.9). Selanjutnya peneliti mencari tahu dari manakah nomor tersebut berasal dan setelah di cari tahu ternyata nomor tersebut berasal dari china (Gambar 4.10). Sehingga dapat diperjelas lagi bahwa sms dengan text "8" ke nomor "1066156686" yang dilakukan *malware android* menyebabkan perangkat *android* berlangganan SMS premium.



Gambar 4.9 Analisa *MessageService.class*

Gambar 4.10 Pengecekan nomor china *http://www.searchyellowdirectory.com/reverse-phone/*

Pada *class* file *CallContentObserver.class* terdapat variabel "*phoneNum",* dimana data nomor telpon dan juga *premium number* termasuk ke dalam variabel "*phoneNum"* (Gambar 4.11). Selanjutnya peneliti melihat kembali *MessageService.class* disana terdapat *"notificationManager"* yang mana dihubungkan dengan *CallContentObserver.class*, sehingga dapat disimpulkan bahwa *"phoneNum"* yang diawali dengan angka "10" maka notifikasi sms masuk atau keluar ditolak sehingga pengguna perangkat *android* tidak mengetahui adanya aktifitas sms yang terjadi (Gambar 4.12)

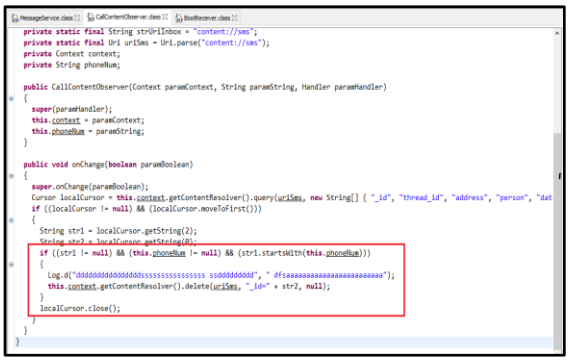
```
public class CallContentObserver
  extends ContentObserver
{
  private static final String strUriInbox = "content://sms";
  private static final Uri uriSms = Uri.parse("content://sms");
  private Context context;
  private String phoneNum;

  public CallContentObserver(Context paramContext, String paramString, Handler paramHandler)
  {
    super(paramHandler);
    this.context = paramContext;
    this.phoneNum = paramString;
  }

  public void onChange(boolean paramBoolean)
  {
    super.onChange(paramBoolean);
```

Gambar 4.11 Analisa *CallContentObserver.class*

```
public IBinder onBind(Intent paramIntent)
{
  return null;
}

public void onCreate()
{
  this.notificationManager = ((NotificationManager)getSystemService("notification"));
  super.onCreate();
  this.sp = getSharedPreferences("sendsms", 3);
  getContentResolver().registerContentObserver(Uri.parse("content://sms"), true, new CallContentObserver(this, "10", null));
}

public void onDestroy()
{
  this.notificationManager.cancel(2131034115);
  this.objHandler.removeCallbacks(this.mTasks);
  super.onDestroy();
}
```

Gambar 4.12 Penolakan notifikasi dari *phoneNum* yang diawali angka "10"

Selain itu, untuk menyempurnakan aktifitas dari *malware android Hippo* SMS supaya tidak disadari oleh *user*, *malware* ini juga memberikan perintah untuk menghapus setiap sms masuk dari china premium *number*. Pada *CallContentObserver.class* telah dijelaskan sebelumnya apabila terdapat sms masuk dari *"phoneNum"* dimana yang diawali dengan angka "10", maka dengan otomatis *malware* ini memberikan perintah *reply* sms ke premium *number* dengan text "ddddddddddsssssssssssssss ssddddddddd" sehingga sms asli yang masuk terhapus (Gambar 4.13). Hal ini menyebabkan pengguna *android* sama sekali tidak mengetahui adanya aktifitas sms premium. Pada aplikasi Hippo (Kuis) ini, aktifitas *malware* tersebut akan aktif setelah terinstall di perangkat *android.*

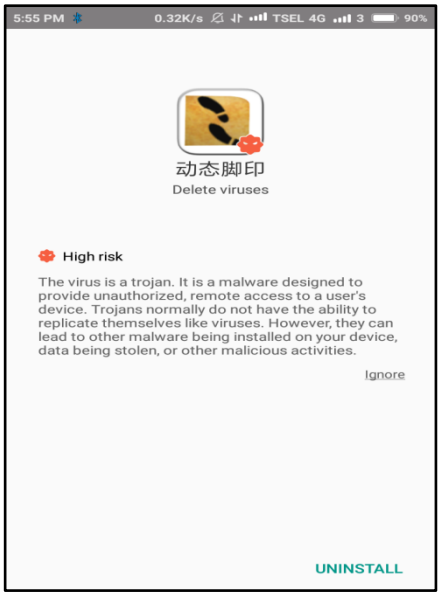


```
private static final String strUriInbox = "content://sms";
private static final Uri uriSms = Uri.parse("content://sms");
private Context context;
private String phoneNum;

public CallContentObserver(Context paramContext, String paramString, Handler paramHandler)
{
  super(paramHandler);
  this.context = paramContext;
  this.phoneNum = paramString;
}

public void onChange(boolean paramBoolean)
{
  super.onChange(paramBoolean);
  Cursor localCursor = this.context.getContentResolver().query(uriSms, new String[] { "_id", "thread_id", "address", "person", "dat
  if ((localCursor != null) && (localCursor.moveToFirst()))
  {
    String str1 = localCursor.getString(2);
    String str2 = localCursor.getString(4);
    if ((str1 != null) && (this.phoneNum != null) && (str1.startsWith(this.phoneNum)))
    {
      Log.d("ddddddddddddddddsssssssssssss ssddddddd", " dfsaaaaaaaaaaaaaaaaaaaaa");
      this.context.getContentResolver().delete(uriSms, "_id=" + str2, null);
    }
    localCursor.close();
  }
}
```

Gambar 4.13 *Script* untuk menghapus sms dari premium *number*

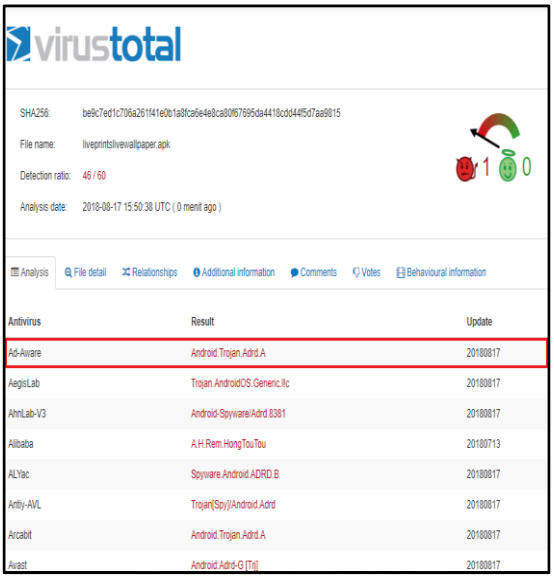
4.2 *Live Prints Live Wallpaper*

Live prints live wallpaper merupakan aplikasi live wallpaper untuk perangkat android dengan tampilan *live prints*, tetapi ternyata aplikasi ini juga mengandung *malware android* yang disebut *Hongtoutou (Trojan.spy.adrd A)*



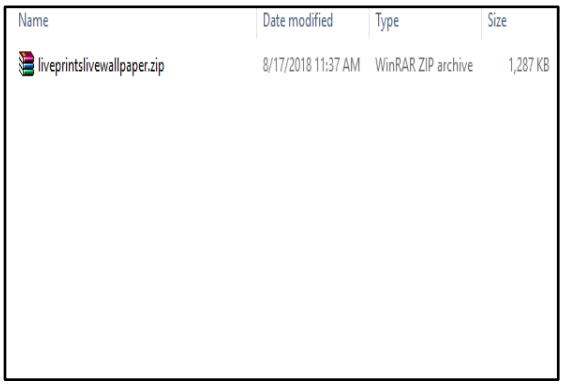
Gambar 4.14 Peringatan dari aplikasi *Live prints live wallpaper.*

Gambar 4.14 merupakan tampilan setelah di install dan terdapat peringatan bahawa aplikasi tersebut mengandung *virus*.



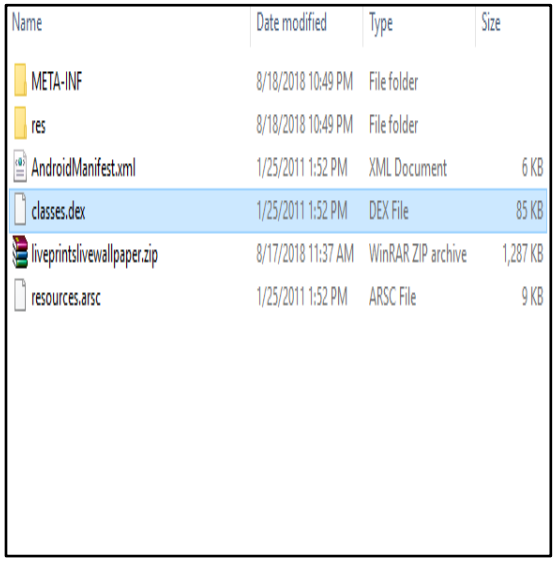
Gambar 4.15 Detail *malware* aplikasi *Live prints live wallpaper*

Langkah selanjutnya adalah merubah ekstensi file *Live prints live wallpaper.apk* menjadi .zip.



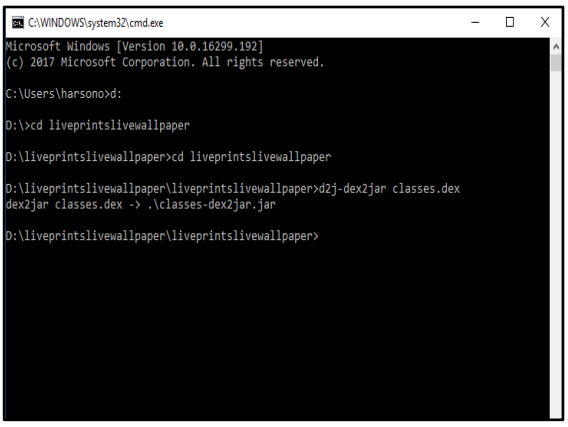
Gambar 4.16 *Rename* file *Live prints live wallpaper.zip*

Kemudian ekstrak file *Live prints live wallpaper.zip* menggunakan *WinRAR* dan akan menghasilkan beberapa file seperti (Gambar 4.17) serta terdapat juga file yang berekstensi DEX.

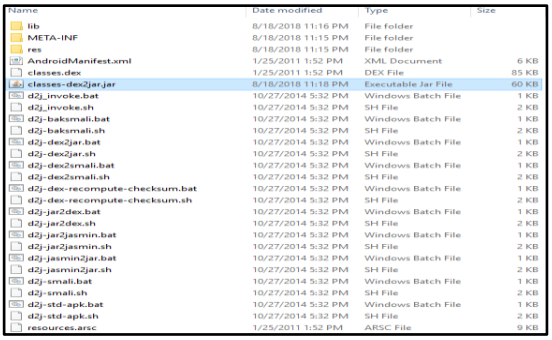


Gambar 4.17 Hasil ekstrak file *Live prints live wallpaper.zip*

Selanjutnya dengan menggunakan *software* dex2jar, file yang berekstensi .DEX di-*convert* menjadi file JAR dengan memanfaatkan jendela *command prompt* (Gambar 4.18) dan menghasilkan file JAR (Gambar 4.19).

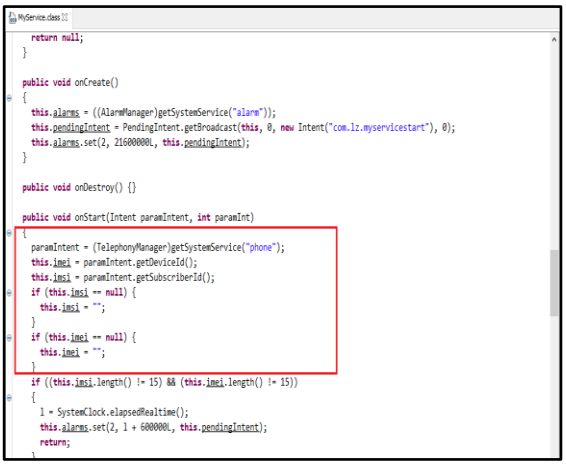


Gambar 4.18 Operasi *convert* dex2jar



Gambar 4.19 Hasil *convert* menggunakan dex2jar

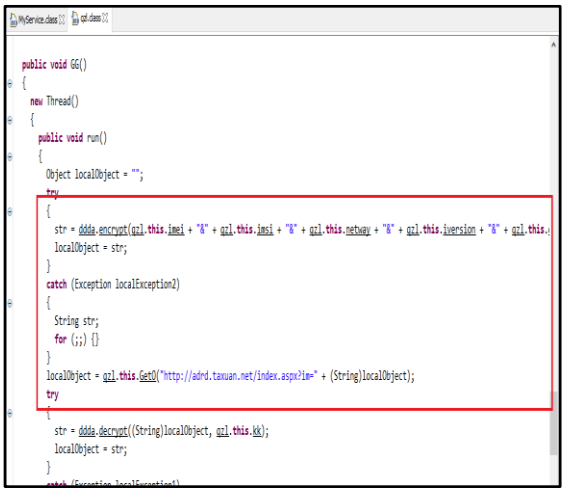
Selanjutnya adalah *decompile* file JAR dengan menggunakan JD-GUI sehingga dapat terlihat keseluruhan *source code java* dari aplikasi *Live prints live wallpaper*.



Gambar 4.20 Perintah untuk melihat *IMEI* dan *IMSI*

Pada Gambar 4.20 terdapat *source code "TelephonyManager"* yang dimaksudkan untuk melihat *IMEI (International Mobile Equipment Identity)* pada perangkat *android* yang terkena *malware*. Setelah *malware* ini membaca *IMEI* dan *IMSI*, selanjutnya *malware* ini mengambil informasi tentang *IMEI* dan *IMSI*.

IMEI dan *IMSI* merupakan idenitas pribadi yang dimiliki setiap perangkat seluler termasuk perangkat *android*, dan tiap perangkat seluler memiliki *IMEI* dan *IMSI* yang berbeda – beda. Informasi yang telah di curi oleh *malware* ini dikirimkan ke situs yang tak dikenal yaitu http://adrd.taxuan.net/index.aspx?im (Gambar 4.21).



Gambar 4.21 Perintah mencuri *IMEI* dan *IMSI* perangkat *android*

Pada aplikasi *Live prints live wallpaper*, *malware* tersebut beroperasi setelah di install dan dijalankan pada perangkat *android*.

5. KESIMPULAN

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

1. Proses analisis *malware* forensik pada aplikasi android dapat dilakukan dengan mengekstrak file apk aplikasi, kemudian mengkonvertnya dengan *tools dex2jar* dan JD-GUI sehingga hasil akhirnya didapatkan *source code java* dari aplikasi tersebut yang nantinya akan di analisis.
2. Aplikasi Hippo (Kuis) terdapat *malware Android.Trojan.HippoSMS.A* yang dapat mengirimkan sms ke salah satu premium *number* china dan mengakibatkan perangkat *android* tersebut berlangganan sms premium dari berbagai premium *number* china tanpa diketahui *user*.
3. Aplikasi *Live prints live wallpaper* mengandung *malware Hongtoutou (Trojan.spy.adrd A)* yang dapat mencuri *IMEI* dan *IMSI* perangkat *android* milik *user* tanpa diketahui.
4. Bagi para pengguna android sebaiknya lebih berhati – hati dalam mendownlaod aplikasi android. Usahakan download aplikasi dari situs yang sudah terpercaya, dan sebelum memutuskan menginstall aplikasi sebaiknya memeriksa terlebih dahulu dengan *antivirus* apakah aplikasi tersebut mengandung *malware*, serta terlebih dahulu memeriksa izin akses yang dijalankan aplikasi.

DAFTAR PUSTAKA

Fediansyah. (2018). Analisis Aktivitas Dan Pola JaringanTerhadap *Eternal Blue*& *Wannacry Ransomware*. Jurnal Sistem Informasi Vol. 02, No.1. ISSN:2460-0921. UIN Raden Fatah Palembang.

Heriyanto, Andri Puspo. 2016. *Mobile Phone Forensics : Theory*. Yogyakarta: Penerbit Andi.

H, Nazaruddin Safaat. 2015. *Android* Pemrograman Aplikasi *Mobile Smartphone* dan *Tablet PC* Berbasis *Android*. Bandung : Informatika Bandung.

Kuswanto, Hendra, Rusmin Syafari dan Timur Dali Purwanto. Analisis Perbandingan Forensik *Recovery* dengan Keamanan Kode Pola dan Kode Kata Sandi pada *Smartphone* berbasis *Android.* Universitas Bina Darma Palembang.

Oktaviani, Resti. Bedjo Sukarmo., dan Nurnawati. Hubungan Iklan *POP-UP Youtube* dengan Sikap *User* (Pengakses). Universitas Surakarta.

Sulianta, Feri. 2016. Komputer Forensik : Melacak Kejahatan Digital. Yogyakarta : Penerbit Andi.