

## Appendix 2

### 1) 神魔龍鬥士 OL-萬神之王

This app contains Tapjoy ads plug-in, which has some malicious behaviors, such as connect to the Internet, send SMS, obtain geo-location information, and obtain phone number. The analysis details are shown below.

#### a) Connect to the Internet

```
URL url = new URL(s);
if(DOWNLOADAPK_TRACE_BUG)
    Log.d("DOWNLOADAPK", (new StringBuilder("getDataSource => myURL = ").append(url).toString());
URLConnection urlconnection = url.openConnection();
urlconnection.connect();
InputStream inputstream = urlconnection.getInputStream();
if(inputstream == null)
    throw new RuntimeException("stream is null");
File file = File.createTempFile(fileNa, (new StringBuilder(".")).append(fileEx).toString());
currentTempFilePath = file.getAbsolutePath();
FileOutputStream fileoutputstream = new FileOutputStream(file);
byte abyte0[] = new byte[128];
```

#### b) Send SMS

```
public void send(Message message)
    throws IOException, InterruptedException
{
    String s = message.getAddress();
    if(message instanceof TextMessage)
    {
        String s1 = ((TextMessage)message).getPayloadText();
        SmsManager.getDefault().sendTextMessage(s, null, s1, null, null);
    } else
    if(message instanceof BinaryMessage)
    {
        byte abyte0[] = ((BinaryMessage)message).getPayloadData();
        SmsManager.getDefault().sendDataMessage(s, null, port, abyte0, null, null);
    } else
    {
        throw new IOException((new StringBuilder("invalid type: ").append(message).toString());
    }
}
```

#### c) Obtain geo-location

```
y = (LocationManager) getSystemService("location");
z = getLocationPrivider(y);
```

```

if(Kernel_trace)
Log.d("onStart location", (new StringBuilder("mLocation01 = ")).append(z).toString());
    if(z != null)
        a(z);

```

#### d) Obtain phone number

```

public static String getPhoneNumber()
{
    return ((TelephonyManager)mainActivity.getSystemService("phone")).getLineNumber();
}

```

## 2) 手機聊天男友

This app contains ooqqx malicious ads plug-in, which has some malicious behaviors, such as connect to the Internet, obtain geo-location, and obtain IMSI code. The analysis details are shown below.

#### a) Connect to the Internet

```

public static boolean getFileByUrl(String s, String s1, boolean flag)
{
    HttpURLConnection httpURLConnection = null;
    httpURLConnection = setupConnection(s, flag);
    httpURLConnection.connect();
    InputStream inputStream = httpURLConnection.getInputStream();
    FileOutputStream fileOutputStream = new FileOutputStream(s1);
    copyStream(inputStream, fileOutputStream);
    inputStream.close();
    fileOutputStream.close();
    boolean flag1;
    if(httpURLConnection != null)
        httpURLConnection.disconnect();
    flag1 = true;
}

```

#### b) Obtain geo-location

```

protected static Location k(Context context)
{
    BufferedReader bufferedReader;
    Location location;
    TelephonyManager telephonyManager;
    bufferedReader = null;
    location = new Location("network");
    telephonyManager = (TelephonyManager)context.getSystemService("phone");
    android.telephony.CellLocation cellLocation = telephonyManager.getCellLocation();
}

```

#### c) Obtain IMSI code

```

public static String o(Context context)
{
    BufferedReader bufferedreader;
    String s1;
    File file;
    bufferedreader = null;
    s1 = ((TelephonyManager) context.getSystemService("phone")).getSubscriberId();
    if(!"mounted".equals(Environment.getExternalStorageState()))
        break MISSING_BLOCK_LABEL_579;
}

```

### 3) 夫妻性愛姿勢雙休

This app contains Vgao and airpush malicious ads plug-in, which has some malicious behaviors, such as connect to a specify website and download data, obtain geo-location, obtain Gmail account information, and obtain IMEI code. The analysis details are shown below.

#### a) Connect to a specify website and download data

```

final String urlString = (new
StringBuilder()).append("https://api.airpush.com/model/user/getappinfo.php?packageName=").append(getPackageNa
me(context1)).toString();
(new Thread(new Runnable() {
    public void run()
    {
        try
        {
            HttpURLConnection httpurlconnection = (HttpURLConnection) (new URL(urlString)).openConnection();
            httpurlconnection.setRequestMethod("GET");
            httpurlconnection.setConnectTimeout(2000);
            httpurlconnection.connect();
            if(httpurlconnection.getResponseCode() == 200)
            {
                StringBuffer stringbuffer = new StringBuffer();
                BufferedReader bufferedreader = new BufferedReader(new
InputStreamReader(httpurlconnection.getInputStream()));
                do
                {
                    String s = bufferedreader.readLine();
                    if(s == null)
                        break;
                    stringbuffer.append(s);
                } while(true);
                Util.jsonstr = stringbuffer.toString();
            }
            httpurlconnection.disconnect();
        }
    }
})

```

```

    }
    catch(MalformedURLException malformedurlexception) { }
    catch(IOException ioexception) { }
    catch(Exception exception1) { }}

```

## b) Obtain geo-location

```

protected static Location k(Context context)
{
    BufferedReader bufferedreader;
    Location location;
    TelephonyManager telephonymanager;
    bufferedreader = null;
    location = new Location("network");
    telephonymanager = (TelephonyManager)context.getSystemService("phone");
    android.telephony.CellLocation celllocation = telephonymanager.getCellLocation();
}

```

## c) Obtain user's Gmail account information

```

static String getEmail(Context context1)
{
    String s = "";
    try
    {
        if(android.os.Build.VERSION.SDK_INT >= 5 &&
context1.checkCallingOrSelfPermission("android.permission.GET_ACCOUNTS") == 0)
            s = AccountManager.get(context1).getAccountsByType("com.google")[0].name;
    }
    catch(Exception exception)
    {
        printLog("No email account found.");
    }
    return s;
}

```

## d) Obtain IMSI code

```

public static String o(Context context)
{
    BufferedReader bufferedreader;
    String s1;
    File file;
    bufferedreader = null;
    s1 = ((TelephonyManager)context.getSystemService("phone")).getSubscriberId();
    if(!"mounted".equals(Environment.getExternalStorageState()))
        break MISSING_BLOCK_LABEL_578;
}

```