

相册木马跟踪分析

Content

[木马信息 3](#_Toc418684654)

[危害程度 3](#_Toc418684655)

[分析步骤 3](#_Toc418684656)

[病毒样本Apk解包 3](#_Toc418684657)

[解包后关键文件确定 4](#_Toc418684658)

[关键文件逐步分析 4](#_Toc418684659)

[strings.xml 4](#_Toc418684660)

[AndroidManifest.xml 5](#_Toc418684661)

[LogcatScannerService.smali 5](#_Toc418684662)

[AndroidLogcatScannerThread.smali 6](#_Toc418684663)

[MainActivity.smali 6](#_Toc418684664)

[MainAPP.smali 8](#_Toc418684665)

[SmsReceiver.smali 9](#_Toc418684666)

[libAPKProtect.so 9](#_Toc418684667)

[分析结果汇总 10](#_Toc418684668)

[病毒工作流程 10](#_Toc418684669)

[病毒主要特点 10](#_Toc418684670)

[病毒对抗分析 10](#_Toc418684671)

[紧急应对方法 11](#_Toc418684672)

[终端用户可以在系统设置中卸载它 11](#_Toc418684673)

[建议各杀毒软件厂商尽快给出查杀办法 11](#_Toc418684674)

[建议运营商可以通过短信的形式发送告警 11](#_Toc418684675)

[威胁情报 11](#_Toc418684676)

[关于绿盟科技 12](#_Toc418684677)

执行摘要

5月4日晚，绿盟科技威胁响应中心监测到在安徽、浙江等区域出现Android聚会相册蠕虫大规模传播的情况，传播速度较快，随即启动应急机制， 应急响应工作随即启动。

1. 5日晨，合肥办接客户获取木反馈，将相关信息提报总部，启动木马分析工作。
2. 5日，对木马样本深入分析，并给出应对方法。
3. 6日，发布紧急通告，并将分析报告发送给客户及合作伙伴。
4. 7日，结合深入分析，更新报告到2.0

如果您需要了解更多信息，请联系：

* 绿盟科技威胁响应中心微博
* <http://weibo.com/threatresponse>
* 绿盟科技微博
* <http://weibo.com/nsfocus>
* 绿盟科技微信号
* 搜索公众号 绿盟科技

木马信息

聚会相册早在2014年就曾经出现过，当时大多是病毒的形式，影响范围有限，在扫描结果中通常可以看到Android.Troj.Sms的标识信息。在2015年劳动节后，绿盟科技威胁响应中心监测到在在安徽、浙江等局部区域，该病毒有大规模传播的趋势，且已经具备蠕虫的特征。该蠕虫传播方式有如下几种：

* 通过感染手机，搜索联系人，发送短信诱使被害人打开恶意网站或者安装APK程序
* 通过感染QQ，搜索联系人，发送链接诱使被害人安装应用程序
* 通过相册APP，进一步扩大发送范围

该蠕虫在传播过程中，可能会发送如下信息：

* 你在干嘛?新年快乐，这是过年时候大家聚会的照片，好珍贵的留影，你快看看吧，地址:http://…
* 这是我们以前的照片你手机打开看看 http://....
* 新年好!这是过年时候大家聚会的照片，好珍贵的留影，你快看看吧，地址：http://...
* 你竟然做出了这样的事!实在让人不能原谅!..../10086.apk;
* 看你干的好事 ，自己看吧…/SS.APK 打開证据都在里面呢;
* 有人悄悄在生日管家记录了您的生日，点击了解http://...
* xxx这是之前的聚会照弄好了，大家的精彩瞬间。手机直接点击地址X安装激活查看…

危害程度

从技术角度上来说，该木马具备简单的反解密措施，以逃避现有杀毒软件的查杀，从目前公开检测报告来看，现有各软件的检测及查杀效果较差[[1]](#footnote-2)，从而对其自我传播形成有利形势。从威胁态势上来看，该木马虽然目前传播后，除消耗话费外并没有做出严重危害动作，但其代码中具备监听手机接收到短信的功能，未来在其大规模感染后，很容利用此功能进行网银、支付宝等金融犯罪动作，其潜在的威胁程度较高。

分析步骤

病毒样本Apk解包

绿盟科技安全人员在获取到病毒样本后，第一时间进行了解包分析，通过使用如下命令可完成 apk的解包

1. $ java –jar apktool.jar d <apkfile\_path>

解包结果如下图所示：



解包后关键文件确定

随即对解包后的各个文件进行分析。在android工程中，排除掉编译时自动生成的文件，以及样本分析关系不大的文件，最终确定以下一些文件为病毒的关键文件：

1. ./res/values/strings.xml
2. ./smali/cn/cnn/mmc/eez/LogcatScannerService.smali
3. ./smali/cn/cnn/mmc/eez/LogcatScannerService$AndroidLogcatScannerThread.smali
4. ./smali/cn/cnn/mmc/eez/MainActivity.smali
5. ./smali/cn/cnn/mmc/eez/MainAPP.smali
6. ./smali/cn/cnn/mmc/eez/SmsReceiver.smali
7. ./AndroidManifest.xml
8. ./lib/armeabi/libAPKProtect.so

其中，

* smali类型的文件，为apk中经过java代码编写后生成的文件；
* Strings.xml为apk中的一些静态字符串变量；
* AndroidManifest.xml为apk开发过程中的配置文件；
* so文件为APP中通过C代码编写的文件。

关键文件逐步分析

对病毒样本的各关键文件的分析如下：

strings.xml

在这个文件中，除了有一个应用名称（聚会相册）外，没有发现有价值的提示信息。



AndroidManifest.xml

这个配置文件中，声明了apk需要的权限，以及一个短信接收监听器(SmsReceiver)。

声明的权限有： 短信接收、短信发送、写联系人、读联系人、读手机状态以及网络访问权限。



LogcatScannerService.smali

该文件代表LogcatScannerService类的声明及实现，在这个类中有4个成员方法，分别为：

Constructor<init>() 、handleLog、onBind 、onStart

其中Constructor为构造函数，无需分析。

**handleLog方法**

经分析，这是一个处理自身类中的静态字符串解码的函数，通过该方法可以将该文件中的密文字符串解码，解码对比如下表所示：

|  |  |
| --- | --- |
| 文件中原密文 | 解码后 |
| s13n8ezq563q7ffm7fet4uD36uztrcfGz8bXxjQ0Ro9DQs4u | android.intent.action.DELETE |
| Q0P06e779vY1QmGYz8z/ | Uinstall |

进一步分析代码结构，得知该代码和android系统的logcat有关，他会监听系统的android.intent.action.DELETE这个消息，并在日志中打印Uinstall这个结果。

**onBind方法**

该函数没有什么特殊的操作，只是返回0值作为函数执行结果，如下图所示：



**onStart方法**

该函数会启动一个名为AndroidLogcatScannerThread的线程，如下图所示：



具体功能可参照本文以下章节。

AndroidLogcatScannerThread.smali

AndroidLogcatScannerThread为 LogcatScannerService类声明的一个内部类，有一个主要方法run函数的实现，同时它又有一同名函数 run(String)，这个函数同handlerLog函数相似，是一个用于解码本类中静态变量的函数，解码结果如下表所示：

|  |  |
| --- | --- |
| 文件中原密文 | 解码后 |
| ozwkICI3QkXNIcLy | logcat |
| 8zVtaWt+ODVeutZm | logcat |
| Q=UNEDIFCBEBWURG2d9GUjEw | waitValue = |

在run的函数体内，做了两件事一个是读日志($ logcat)，另一个是其清空日志（$ logcat -c）

下图为调用Runtime.exec(“logcat”).waitFor() 的代码片段：



经过更细致的分析，发现这个文件的代码和网上流传的代码完全相同[[2]](#footnote-3)（除静态代码做了加密处理）。这是一段监听自身被卸载的代码，但从样本分细结果看，该代码除了往日志里打了一个Uinstall的消息外，无任何后续操作。

MainActivity.smali

这是该样本的核心功能文件，其所对应的MainActivity这个类中有几个重要的方法:: onCreate、isInstallation、readContacts.

如下对各个方法进行详细分析：

**onCreate**

由于这个类是窗口类，此时onCreate是继承自Activity的重写方法。在该方法中配置了一个WebView控件，应用打开时看到的QQ空间界面就是通过这个WebView展示的，如下图所示：



在该类中还实现了一个onKeyDown方法，用来关联“实体按键”和“WebView的响应逻辑”，在测试机中，按动物理按键会导致程序崩溃，随后应用图标消失。

**sInstallation**

该函数是一个判断运行环境的函数，和恶意功能关系不大。值得注意的是该类中还存在一个同名函数，如下图所示：

 

该函数同HandleLog相同，也是硬编码字符串解密的函数，本类中的几个字符串解码结果如下所示：

|  |  |
| --- | --- |
| 文件中原密文 | 解码后 |
| Mz+ChZCdnZCFmJ6fQ0HXbGuJ | Installation |
| ==E8Oy4jIy47JiAhMjiupr00NABi | Installation |
| U=1wd2Jvb2J3amxtREOjXLBO2jSm | Installation |
| Q=6/pbCypY64tfHs8UFBpZfETzsr | contact\_id = |
| M0O2o/M5QsBjkVpq | data1 |
| ==21ur2q4kIxLqyDMwtq  | number: |
| s1bSn8PIn+Lhn+vYlcblnPLrnO/Oner8nsD8nsL6nsDhnsHfn/P3neD+nNPbnfP9lcb2nObzndPAktTKn8Ttnebxnebxn+nclcb2n/n1nfP9n+bKn+f6Wg5UGRRVKDtLIDgoSTUziSSEznn+ | 在干嘛呢？我整理了一些以前的橡片，有空记得看看哦，像片地址 t.cn/RA1ZBR3 |

**readContacts**

这是和本样本传播有关的代码，它的核心功能有两项，遍历本机联系人及发送短信：

* 就是遍历本机联系人（通讯录）：
	+ 
* 然后分别给他们发特定内容的短信（比如，在干嘛呢….）：
	+ 

MainAPP.smali

这是调用反调试库的类，代码就是加载libAPKProtect.so文件，没有其他功能，如下图所示：



SmsReceiver.smali

该类是一个恶意的收发短信的代码。

在这个类中有一些硬编码的字符串是加密的，解密对照表如下所示：

|  |  |
| --- | --- |
| 解密前 | 解密后 |
| oziClImPgsiWlImQj4KDlMiyg4qDlo6JiJ/Itau1ubSjpaOvsKOiNESMZnh4 | android.provider.Telephony.SMS\_RECEIVED |
| AztJAFtFAnlBDGNOC1h+NELe06A3 | 短信来自： |
| 8zPxuOP9udrZufLls+DGQzN6y6u8 | 短信内容： |
| I1OTk8enp8eOjsqiotCHh9CZmTUxqJC6q3k5 | yyyy-MM-dd HH:mm:ss |
| 01PFxsTGxsvDwcDLwjVBxLJoQEws | 1065755802381 |
| c1heXV9dXVBYWltQWUNGbqNQlHWV | 1065755802381 |
| k1BWVVdVVVhQUlNYUUM3syCNlnUV | 1065755802381 |

在该类中实现了一个onReceive函数，这个函数有一个恶意发短信的功能，逻辑如下：

当被感染机器收到新短信时，会按照一个新格式来组织短信内容，并发到1065755802381这个号码。

其中短信的组织格式为”短信来自：xxxxxxxxx 短信内容：xxxxxxxxxxx <yyyy-MM-dd HH:mm:ss>”

libAPKProtect.so

这个库文件是经过C语言编写完成的，JNI\_Onload是其主函数，函数中用到了ptrace来修改程序自身的一些数据。经IDA简单分析后的截图如下：



分析结果汇总

从如上分析中可以看到，这个蠕虫由于利用了地址进行信息分发，其诱惑性及成功几率极高，而且在传播过程中由于群发短信，将会对受害者产生额外的手机话费消耗，具体结果如下：

病毒工作流程

* 搜索电话号码本的每个号码，并向其发送短信。
* 短信内容为：“在干嘛呢？我整理了一些以前的橡片，有空记得看看哦，像片地址 t.cn/RA1ZBR3”
* 当短信接收者收到这条短信，并打开那个短地址后，会继续下载到新的apk程序，并在被安装机器上继续执行步骤1、2的流程。
* 在被感染机上监听接收短信的状态，当收到新短信时会向1065755802381这个号码转发这条短信内容。从百度搜索结果看，1065xx这个号码可能是一个腾讯废弃的服务号码，这里没有联系腾讯的员工做进一步确认，测试者向其发送短信也很少发送成功。

病毒主要特点

* 通过社交网络（电话本）以及联系人间的信任程度（相信发信人随后打开信息中的链接）进行传播。
* 后台自动发送短信，消耗被安装机器的话费余额。
* 在样本中发现其存在监听手机短信的功能。虽1065755802381这个号码可能已被废弃（测试者向其发送短信无法发送成功），但仍应考虑其存在窃取短信的可能，如发现被该样本感染的机器应当注意短信被窃取的风险。

病毒对抗分析

由于要躲避查杀，该蠕虫在对抗分析上做了很多的工作。

* 程序中的特征字符串均已加密，要经过动态解密才能还原。
* 样本中附带so文件，以ptrace方式还原程序自身的部分程序片段，达到对抗分析和反调试的效果。
* 程序运行中会执行 logcat –c 这个命令，来清除自身运行过程中产生的日志，令自身不容易被察觉。
* 在测试机中，运行一次后程序图标就会自动消失，但程序本身并未消失。需要手动卸载才能清除。

紧急应对方法

终端用户可以在系统设置中卸载它

在android手机的系统设置中，依次点击：应用程序管理应用程序聚会相册，来选定蠕虫文件，随后在新的窗口中点击卸载按钮，并确认，即可清除。如下图所示：



建议各杀毒软件厂商尽快给出查杀办法

建议各杀毒厂商尽快结合如上分析，更新查杀机制or病毒特征码，以便尽快实现该病毒的查杀，进而争取在其大范围传播之前控制住这个态势。

建议运营商可以通过短信的形式发送告警

在之前类似的案例中，各运营商可以采用短信的形式向用户直接发送告警短信，将病毒出现的特征解释清楚，效果还是不错的。同时，运营商可以采取措施拦截病毒短信，全面封堵下载地址。

威胁情报

从此次相册木马蠕虫的传播情况可以看到，无论其原理怎样，无论防护方案如何实施，关键在于尽可能快的了解到漏洞信息及相关的情报，以便尽可能快的启动应急响应机制。这无论对于解决传统安全或者APT攻击来说都是重要的手段之一，威胁情报的获取及响应都体现了防御能力的建设程度，威胁情报服务体系至少包含了威胁监测及响应、数据分析及整理、业务情报及交付、风险评估及咨询、安全托管及应用等各个方面，涉及研究、产品、服务、运营及营销的各个环节，绿盟科技通过研究、云端、产品、服务等立体的应急响应体系，向企业和组织及时提供威胁情报并持续进行后续服务，保障客户业务的顺畅运行。

如果您对我们提供的内容有任何疑问，或者需要了解更多的信息，可以随时通过在微博、微信中搜索绿盟科技联系我们，欢迎您的垂询！

关于绿盟科技



北京神州绿盟信息安全科技股份有限公司（简称[绿盟科技](http://www.nsfocus.com/)）成立于2000年4月，总部位于北京。在国内外设有30多个分支机构，为政府、运营商、金融、能源、互联网以及教育、医疗等行业用户，提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。

基于多年的安全攻防研究，绿盟科技在网络及终端安全、互联网基础安全、合规及安全管理等领域，为客户提供入侵检测/防护、抗拒绝服务攻击、远程安全评估以及Web安全防护等产品以及专业安全服务。

北京神州绿盟信息安全科技股份有限公司于2014年1月29日起在深圳证券交易所创业板上市交易，股票简称：绿盟科技，股票代码：300369。

1. VirSCAN：http://r.virscan.org/report/4cb994392b1fc3c3d02011fa3dbbd49f [↑](#footnote-ref-2)
2. http://blog.csdn.net/pi9nc/article/details/23163151 [↑](#footnote-ref-3)