昆明学院学报　2016,38(6):56~62　ISSN 1674-5639　CN 53-1211/G4
Journal of Kunming University

Android 手机木马提取与分析

赵　鑫[1],郭红怡[2],杨　晶[3]
(1. 甘肃省公安厅 网络安全保卫总队,甘肃 兰州 730030;2. 云南警官学院 信息网络安全学院,云南 昆明 650223;
3. 中国人民公安大学 网络安全保卫学院,北京 大兴 102623)

摘要:随着手机网民的急剧上升,电信诈骗产业链迅速蔓延,诈骗者依靠伪基站设备发送钓鱼网址,给用户手机植入 Android 木马,在用户不知情的情况下完成远程转账. 该电信诈骗方式具有丰富的攻击方法,包括隐私窃取、短信拦截、远程控制和钓鱼等,而网络案件应通过网络的思维方式进行侦查,因此要求侦查人员必须掌握网络钓鱼的原理、Android 木马反编译、应用程序安装包提取等专业知识. 基于上述分析,通过案例探讨在电信钓鱼诈骗案件中,Android 木马 apk 的提取方式和 apk 反编译分析,并通过示例操作进一步认识 Android 手机木马的提取和分析方法.

关键词:网络诈骗;手机木马;adb;反编译
中图分类号:TP393　**文献标识码**:A　**文章编号**:1674-5639(2016)06-0056-07
DOI:10.14091/j.cnki.kmxyxb.2016.06.013

Detecting and Analyzing on Android Cell Phone Trojan

ZHAO Xin[1],GUO Hongyi[2],YANG Jing[3]
(1. Network Security Guard,Gansu Province Public Security Department,Lanzhou,Gansu,China　730030;
2. School of Information and Network Security,Yunnan Police College,Kunming,Yunnan,China　650223;
3. College of Information Technology and Network Security,People's Public Security University of China,Daxing,Beijing,China　102623)

Abstract: Along with the sharp increasing of mobile Internet users, telecommunications fraud industry chain spreads quickly. Relying on pseudo base station equipment, the fraudsters send phishing site, implant Trojans to Android, complete remote transfer accounts without knowing of the users. This kind of telecommunications fraud methods enriches the attacking methods, including filching privacy, intercepting message, remote control, and fishing, etc. Internet cases must be investigated through network thinking and the investigators must understand and master the principles of phishing, Android Trojan decompilation, application installation package extraction and other professional knowledge. Based on the analysis above, we studied the cases of phishing scams in which Android Trojan apk extraction method and the apk decompilation were analyzed, and through the example to know the methods of Android Trojan detection and analysis.

Key words: telecommunications fraud;cell phone trojan;adb;decompilation

网络诈骗是以手机等可上网的设备为媒介,采用电信设备通过欺骗的方式骗取款额较大的公私财物的犯罪活动,其中网络钓鱼的危害较大[1]. 诈骗团伙的诈骗方式是使用伪基站设备进行钓鱼短信群发,利用被害人好奇、贪财、好利的心理,诱骗被害人点击短信中的钓鱼链接,被害人只要点击了犯罪团伙提供的钓鱼链接,其手机就将被安装木马,诈骗团伙能够实现远程控制,达到盗取被害人手机银行中钱财的目的. 整个诈骗犯罪链:木马开发、钓鱼短信发送、钓鱼网站制作、远程控制"偷钱",每一环节都由诈骗团伙的专人负责. 而由于受到侦查意识和专业技术的限制,侦查人员通常情况下会从资金流转入手侦查,往往只能抓获负责取钱的马仔,很难从木马回传地址、钓鱼网站地址等多个侦查入口出发,实现从源头快速精准打击犯罪的效果.

收稿日期:2016-10-31
作者简介:赵鑫(1990—),男,甘肃陇西人,网络工程师,主要从事网络安全与计算机犯罪侦查研究.

本文主要从专业技术的角度出发,提取和分析 Android 木马,为侦查此类案件提供新的思路,从而实现快速提取出有价值线索,抓获犯罪主要嫌疑人,挽回受害人损失,有效打击网络诈骗犯罪的目的.

1　Android 手机木马提取

1.1　常见电信钓鱼诈骗方式

犯罪团伙通常会以伪装的毕业相册、中奖、改签机票等信息为名,通过伪基站、voip 和 170 等虚拟号码将虚假信息发送给被害人,诱导被害人点击短信中的木马链接,然后通过安装的木马软件,窃取被害人的银行卡信息,拦截短信验证码,盗取被害人的财产.

1.2　提取手机木马的几种方式

1)用浏览器直接下载. 通常犯罪团伙会以短信的形式,加一段欺骗性、诱惑性的语言让被害人点击其提供的网址,当被害人点击这个链接时手机会自动下载并安装远程控制木马. 如果被害人能够及时提供诈骗短信内容,特别是信息中的钓鱼网址,侦查人员则可以通过电脑浏览器输入钓鱼网址下载诈骗团伙用于作案的手机木马.

2)被害人手机下载安装包提取. 被害人手机在安装木马时通常会有一个安装包下载到手机本地,其可能具有安装之后自动销毁的功能. 对于这种情况,侦查人员可使用手机取证软件对被害人手机做数据恢复,提取已被删除的安装包文件,导出文件并提取相应的木马安装包文件.

3)用 adb 命令通过安装路径提取[2]. ADB 是客户端/服务器端管理程序,其中客户端是用来操作的终端电脑,服务器端是 Android 系统的设备. 在电脑上安装客户端,在手机上打开 USB 调试模式就可以完成. 在 Windows 的控制端口下通过以下命令可以查看安装的 adb 及环境变量设置是否成功,如图 1 所示.

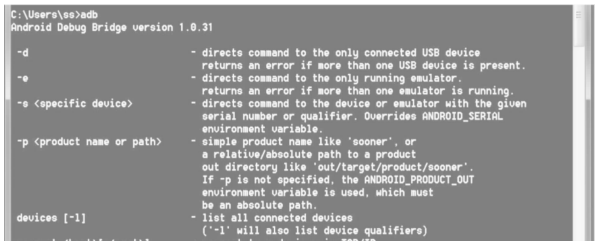


图1　adb查看参数

通过以上命令可以看出安装的 adb 版本和常用命令参数. 打开被害人手机的 USB 调试模式,在 adb 下使用命令查看设备是否正常连接访问,如图 2.

图2　adb查看设备命令

这个命令查看当前连接到计算机的设备,则会通过相应的命名方式显示.

设备正常连接,这样可以通过命令来查看手机中安装的应用程序列表,找到目标远程控制木马程序的安装包路径,如图 3 和图 4.

图3　列出应用程序安装包

图4　目标程序的安装包路径

通过以上命令,已经找到了目标远程控制木马程序的安装包名称,然后通过下列命令可以设置目标远程木马安装数据路径,并将程序重新打包成安装包复制到客户端电脑,如图 5.

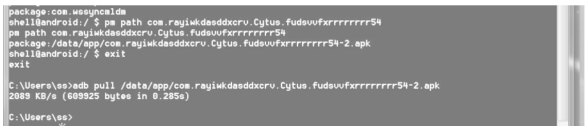


图5　将文件复制到电脑

以上命令已经将目标远程木马程序复制到本地文件,并使用原始名称,保存到默认根目录下,如图 6.

4)手机安全管家提取. 打开手机安全软件,通过数据线使被害人手机正常连接电脑,然后在已安装的程序中找到诈骗团 + 伙远程控制木马的应用程序,手机安全管家提供程序导出功能,如图 7.

通过手机管家也可以对手机中安装的远程木马控制程序进行提取.

5)将 apk 下载地址复制到迅雷下载中,如果之前有人通过迅雷下载过此木马程序,迅雷会在服务器上存储 apk 备份文件.

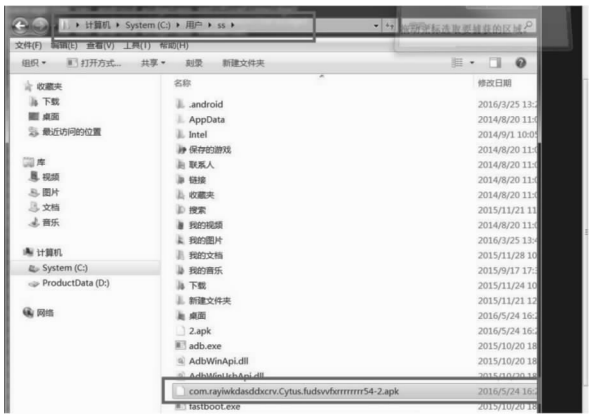


图6 导出文件查看

图7 手机安全管家导出apk

6)通过技术手段查询与此网站域名相似的网站,分析相似网站域名,查看是否存在其他域名可以下载此木马程序.

1.3 Android 木马固定

通过对以上提取的文件进行分析,两文件名称各为“资料. apk”和“com. rayiwkdasddxcrv. Cytus. fudsvvfxrrrrrrrr54-2. apk”,其

包名:com. rayiwkdasddxcrv. Cytus. fudsvvfxrrrrrrr54

文件类型:APK

SHA1:

72c5b229f27e761a3b315ef500c9da0a5c5bac6a

MD5:eb3d2a5b6b68688cd57e70fd2bb2934d[virustotal]

SHA-256:

45b8cf3dc1c0c5a26f34d2a282fadd53015e425acc3e27a3dfbd49410ad007f0

文件大小:0.58 MB

但是,被害人通常的做法是对自己手机进行刷机或者还原出厂设置,这样提取到 APK 文件的可能性很小,因钓鱼网站的有效期很短,得不到及时的提取链接也将失效,这样就会给侦查工作带来严重的不便.

2 Android 应用程序分析

2.1 分析原理

Android 系统是由安迪·鲁宾(Andy Rubin)等最早研发,系统最初的目的是应用于数码相机,因市场需求不大,经不断地改造升级,于是向智能手机的操作系统转换. 而 Android 的源代码是开放的,Android 操作系统具有很强的开放性和可移植性,因此该系统被广泛应用于很多平台和电子设备[3].

APK 是 AndroidPackage 的缩写,即 Android 安装包(apk),APK 文件其实是 zip 格式[4],但后缀名被修改为 apk,其压缩包中的部分文件夹可选,比如 assets 文件夹是存放资源,该文件夹可选;lib 文件夹是存放编译好的 so 文件,该文件夹也是可选,so 文件类似于 dll,里面存放被调用的函数;res 文件夹是存放资源文件的目录. META-INF 文件夹中包含 3 个文件:MANIFEST. MF, CERT. SF 和 CERT. RSA, MANIFEST. MF 是除 META-INF 目录之外的所有文件的 Message Digest 信息添加到该文件中,这些信息与 APK 包中的文件实体是一一对应的,在该文件中的信息并没有用密钥进行加密,而只是各个文件的 Message Digest 信息, CERT. SF 是一个 MANIFEST. MF 文件的被签名版本,通过该文件,可以察觉任何的增加未签名文件的操作,签名文件在 SHA1-Digest-Manifest 中包含了 MANIFEST. MF 文件的 message digest,此外还包含了在 MANIFEST. MF 文件所列出的各个文件的 message digest. CERT. RSA 与 MANIFEST. MF 文件的构成很相似,只不过签名文件中的 message digest 是基于 MANIFEST. MF 文件中的内容进行计算的,而 MANIFEST. MF 文件中的 message digest 是基于 APK 包中的文件内容本身进行计算的. AndroidManifest. xml 是程序全局配置文件.

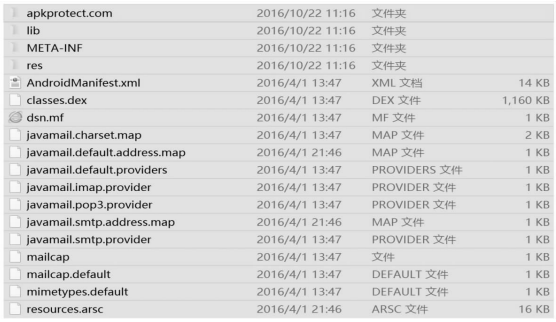
本次案例是用 adb 和手机安全管家提取到的 apk,通过反编译软件对受害人手机中提取的 apk 木马进行分析,得到控制端使用的回传手机号码和邮箱账号、密码及木马编写作者信息,为案件侦办提供了有力的线索和证据.

2.2 动态分析

修改文件后缀为 zip,打开文件查看文件所包含内容,如下图 8.

从以上 apk 文件可以看出,木马是通过邮箱回

传信息. 以上文件包含了网络管理协议 IMAP、邮件协议 po 和 smtp 文件.



apkprotect.com	2016/10/22 11:16	文件夹	
lib	2016/10/22 11:16	文件夹	
META-INF	2016/10/22 11:16	文件夹	
res	2016/10/22 11:16	文件夹	
AndroidManifest.xml	2016/4/1 13:47	XML 文档	14 KB
classes.dex	2016/4/1 13:47	DEX 文件	1,160 KB
dsn.mf	2016/4/1 13:47	MF 文件	1 KB
javamail.charset.map	2016/4/1 13:47	MAP 文件	2 KB
javamail.default.address.map	2016/4/1 21:46	MAP 文件	1 KB
javamail.default.providers	2016/4/1 13:47	PROVIDERS 文件	1 KB
javamail.imap.provider	2016/4/1 13:47	PROVIDER 文件	1 KB
javamail.pop3.provider	2016/4/1 13:47	PROVIDER 文件	1 KB
javamail.smtp.address.map	2016/4/1 21:46	MAP 文件	1 KB
javamail.smtp.provider	2016/4/1 13:47	PROVIDER 文件	1 KB
mailcap	2016/4/1 13:47	文件	1 KB
mailcap.default	2016/4/1 13:47	DEFAULT 文件	1 KB
mimetypes.default	2016/4/1 13:47	DEFAULT 文件	1 KB
resources.arsc	2016/4/1 21:46	ARSC 文件	16 KB

图8　apk文件包含文件

通过分析 APK 程序中 META-INF 文件下的 Manifest 文件,可以得出 APK 文件的权限如图 9.

```
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.RECEIVE_USER_PRESENT" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.BROADCAST_SMS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.WRITE_SMS" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_LOGS" />
```

图9　Manifest文件中的权限部分

Activity 是可以显示控件的单独屏幕,并且可以对用户的事件做出响应,不同的 Activity 之间使用 Intent 协议进行通信. Intent 有两个最重要的部分:Action 和 Action 对应的数据,常见的 Action 有 Main,DELETE,VIEW. 分析 Manifest 文件的 Activity 组件如下图 10 和图 11.

```
<activity android:label="@string/app_name"
    android:name="cn.android.emial.MainActivity"
    android:excludeFromRecents="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

图10　Action的MAIN

```
<activity android:label="卸载程序" android:icon="@drawable/ic_launcher"
    android:name="cn.android.emial.UninstallerActivity">
    <intent-filter android:priority="2147483647">
        <action android:name="android.intent.action.VIEW" />
        <action android:name="android.intent.action.DELETE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="package" />
    </intent-filter>
</activity>
```

图11　Action的view，delete，default

从以上分析可以看出:android. intent. action. MAIN 决定应用程序最先启动的 Activity;android. intent. action. DELETE 决定应用程序启动之后即删除;android. intent. category. LAUNCHER 和 android. intent. action. VIEW 决定应用程序是否显示在程序列表里和图标是否隐藏.

Service 组件是由 startService( )方法启动的服务,它和调用者之间没有任何关系,即使调用者关闭,Service 服务仍可运行,停止 Service 服务是通过调用 Context. stopService( )来实现,紧接着系统会调用 onDestory( ). 如果 Service 服务是首次启动,系统会用先后调用服务 onCreate( ) - - > onStart( )的方法;如果 Service 服务已经启动,系统调用只会采用触发 onStart( )的方法;bindService( )是启动服务与调用者绑定的方法,二者同步进行,调用者的关闭将导致服务的终止;此方法首次启动服务时,系统会先后调用服务的 onCreate( ) - - > onBind( ),假如服务已启动,系统不会再触发前面 2 个方法;如果调用者已退出,系统会先后调用以下 2 个方法 onUnbind( ) - - > onDestory( ). 系统可使用 Contex. unbindService( )方法,依次调用方法 onUnbind( ) - - > onDestory( ),可主动解除绑定. 分析 Manifest 文件的 Service 组件,如图 12.

```
<service android:name="cn.android.emial.SmSserver" android:enabled="true" />
```

图12 Service文件

Receive 组件是 BroadcastReceive,没有用户界面,通过接受外部事件的广播启动. BroadcastReceive 可以启动一个 activity 或 serice 来响应它们收到的信息,如图 13.

手机 apk 中,Broadcast 组件是不可缺少的一部分,其主要有两种类型:1)普通广播,其是通过 Context. sendBroadcast(Intent myIntent)方法发送的 BroadcastReceive 来启动 Activity 或 Serice 组件来响应收到的传播信息[5];2)有序广播,其是通过 Context. sendOrderedBroadcast (intent, receiverPermission)方法发送的,广播的优先级是第 2 个参数决定,优先级随数值的增大而越高(数值是在 -1 000 ~1 000 之间),接收广播时的优先级可通过 intentfilter 中的 priority 进行设置,当值设为 2 147 483 647 时优先级最高. 接收顺序为同级别接收的先后顺序是随机的,高级别比低级别先接收,先接收到则可通过 abortBroadcast()方法截断其他接收者接收该广播.

```
<receiver android:name="cn.android.emial.BootReceiver">
    <intent-filter android:priority="2147483647">
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
        <action android:name="android.provider.Telephony.SMS_RECEIVED_2" />
        <action android:name="android.provider.Telephony.GSM_SMS_RECEIVED" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
    <intent-filter android:priority="2147483647">
        <action android:name="android.intent.action.PACKAGE_RESTARTED" />
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <action android:name="android.intent.action.USER_PRESENT" />
    </intent-filter>
</receiver>
```

图13 BroadcastReceive工作信息

2.3 静态分析

本文的 apk 分析平台是在 Windows 10 专业版下,系统类型 32 位,处理器为 intel(R)core(TM)i7 CPU 920@ 2,67 GHZ. 安装 java 环境,修改环境变量,配置 AndroidKiller-v1. 3. 1.

使用 AndroidKiller 对资料 . apk 进行反编译,AndroidKiller 提示 apk 已加固,点击 OK 解析,如图 14.

点击 OK,尝试去编译,编译成功,AndroidKiller 稍微智能一些,它根据函数对当前反编译代码会做一个初步的分析,标记程序的权限和功能,如图 15.

图14 AndroidKiller反编译时的提示信息

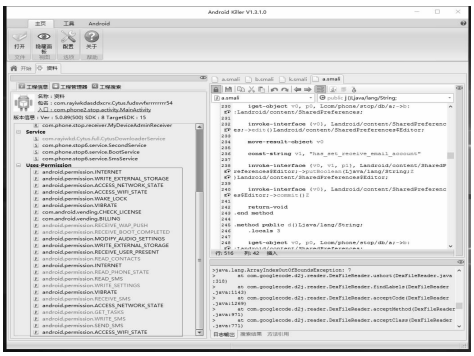


图15 AndroidKiller标记的权限和功能

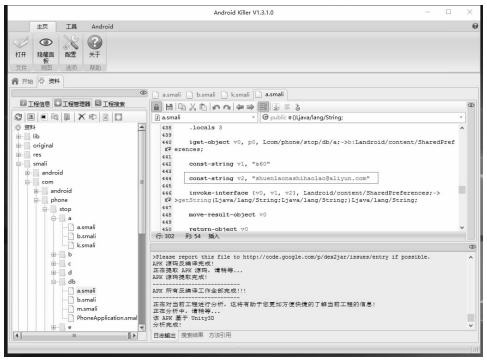


图16 回传邮箱账号

通过分析木马作者设置的拦截信息回传邮箱账号、密码、手机号、木马标识特征码[6],如图 16、图 17 和图 18.

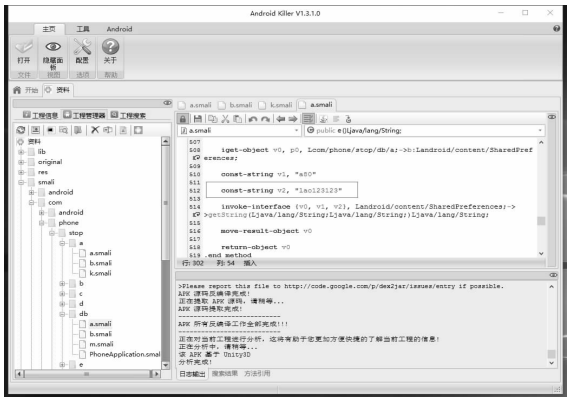


图17　回传邮箱密码

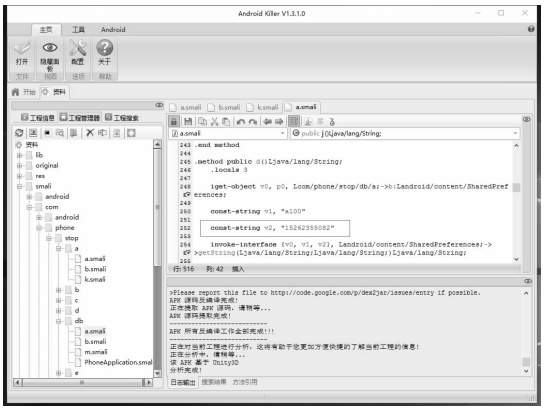


图18　回传手机号

通过邮箱账号尝试登陆邮箱,但因当时该案发已经有月余,账号通过多方举报或后台检测有恶意行为,现已被冻结,无法登陆,要想知道邮箱中收取的邮件内容只能通过其他方法调取. 如果能及时地完成以上分析工作,将为案件的侦办提供有效的线索,从而进行查证.

3　移动恶意程序现状

2016 年第二季度移动端恶意程序新增量和感染量统计数据显示[7](如图 19),新增样本量呈递增趋势,6 月份最多达到 174.8 万个;而感染量呈下降趋势,6 月份达到最低,为 1 933 万次.

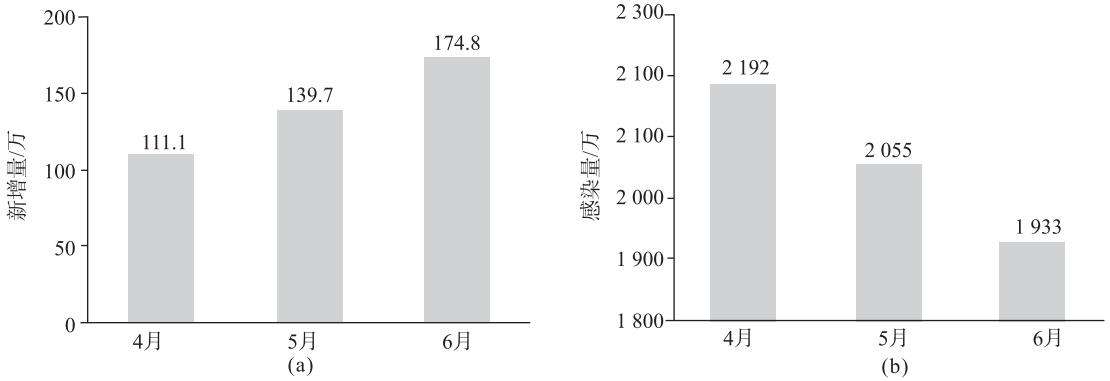


图19　2016年第二季度移动端恶意程序新增量和感染量统计

手机端网络诈骗分类情况统计表明,手机端网络诈骗有下降趋势. 但是虚假兼职虽然相比 2015 年第二季度的 22.6% 有所下降,仍然以 20.2% 高居首位,身份冒充 16.6% 紧随其后,虚假购物以 11.9% 位居第 3[8],这 3 个诈骗团伙在网络诈骗界经久不衰,统治力依旧强劲. 如图 20.

以上数据分析显示,随着人们安全意识和打击力度的不断提高,网络诈骗虽有下降趋势,但诈骗团伙仍在与受害者、安全部门斗智斗勇,该团伙广泛利用高科技手段,不断调整运行模式,并出现了各种各样的诈骗方式. 其总体模式是网络电信诈骗集团通过搭建网络钓鱼平台,并利用技术手段种植手机端控制程序,窃取手机中的个人信息,并利用各种网络技术,快速地将受害者的资产转移、套现,躲避对其账户的冻结. 而电子数据最大的特性就是易于毁坏和丢失,因此对电子数据勘察取证成为侦查员不可缺少的技能,案件发生时,应及时收集、固定电子数据并对电子数据进行分析,快速找到侦查线索,为案件的侦办提供正确线索和侦办思路.

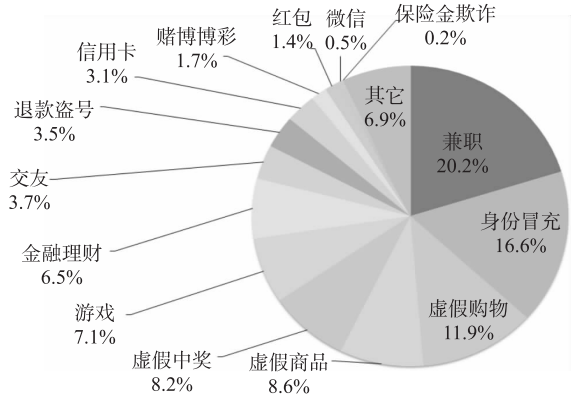


图20 2016年第二季度手机端网络诈骗分类情况

4 结论

网络案件的侦办离不开网络技术和网上线索，本文主要是从网络案件一般的侦办思路，以及网络线索出发，利用技术手段介绍了提取、固定、分析手机 apk 文件的一般原理和方法，为案件侦办提供线索和证据.

随着互联网的快速发展，各行各业已离不开网络，各类案件的侦办也是如此，因此通过网络手段和线索对案件进行侦办也是不可缺少的方式，及时收集、固定、提取、分析网络数据和电子证据将对案件的侦办有着非常重要的意义. 网络没有通常意义的地域之分，改变了通常的通讯方式. 在网络案件侦办过程中，快速整合网络资源，有效地利用网络技术，对案件线索进行及时分析固定，查找电子线索，应用网络原理还原案件过程，可以提高破案效率.

[参考文献]

[1]程建，朱赟，王春丽. 对短信诈骗犯罪定性与证据审查的思考[J]. 法制与社会，2010(26)：135 - 136.

[2]杨峻. Android 系统安全和反编译实战[M]. 北京：人民邮电出版社，2015.

[3]丰生强. Android 软件安全与逆向分析[M]. 北京：人民邮电出版社，2013.

[4]张博. 网络诈骗犯罪侦办的困境与对策研究[D]. 大连：大连海事大学，2015.

[5]彭国军，邵玉如，王泰格. 基于 Android 的手机隐私保护技术及实现[J]. 信息网络安全，2012(4)：54 - 57.

[6]贾菲，刘威. 基于 Android 平台恶意代码逆向分析技术的研究[J]. 信息网络安全，2012(4)：61 - 63.

[7]诸葛建伟. 安卓手机系统的安全威胁及应对[J]. 中国信息安全，2013(8)：77 - 79.

[8]ZHAO Y J，WANG Z，ZHOU W，et al. Hey，you，get off of my market：Detecting malicious apps in official and alternative android markets[C]//Proc of the 19th Network and Distributed System Security Symposium，San Diego，2012：712 - 725.