

# 基于 Android 平台的手机木马的设计与实现

吴玮斌<sup>1</sup>, 孟 魁<sup>1</sup>, 徐 林<sup>2</sup>, 刘功申<sup>1</sup>

(1. 上海交通大学 电子信息与电气工程学院, 上海 200240;

2. 公安部第三研究所信息安全公安部重点实验室, 上海 200120)

**摘 要:** Android 作为三大智能手机操作系统之一, 由于其是开源性系统, 因此成为了黑客攻击的重要目标。而木马作为一种隐蔽性、欺骗性很高的攻击手段, 正在该平台上不断蔓延, 虽然已经受到了广泛关注, 但却没有很好的抑制方式。文中设计并实现了一种简单的木马原型。通过对命令接收, 信息获取, 数据回传, 远程控制四个 Android 木马的主要功能分析, 提供了一种实现思路, 并且详细阐述了该平台上各种信息获取方式, 借此探讨 Android 系统的脆弱性, 并期望能提出针对这种木马的防范措施。

**关键词:** Android; 手机木马; 信息获取; 远程控制

**中图分类号:** TP393.08

**文献标识码:** A

**文章编号:** 1673-629X(2014)10-0155-04

doi:10.3969/j.issn.1673-629X.2014.10.037

## Design and Realization of Mobile Trojan Based on Android Platform

WU Wei-bin<sup>1</sup>, MENG Kui<sup>1</sup>, XU Lin<sup>2</sup>, LIU Gong-shen<sup>1</sup>

(1. School of Electronic Information and Electrical Engineering, Shanghai Jiaotong

University, Shanghai 200240, China;

2. Key Lab of Information Network Security of the Third Research Institute of Ministry of Public Security, Shanghai 200120, China)

**Abstract:** As one of the three major smart-phone system, Android has already become the attack target of the hackers because of its open source. Mobile Trojans is an elusive and fraudulence attack method and is becoming wide spread on this platform. Although a lot of attention is paid on it, but do not have good solution to it. In this paper, design a simple Trojan which realizes command receiver, information acquisition, data transfer and remote control, elaborating the various ways to obtain information on Android platform aiming to discuss the weakness of Android system and acquire a defensive method of the Trojan.

**Key words:** Android; mobile Trojans; information acquisition; remote control

## 0 引 言

随着科技的发展, 智能手机与各类平板电脑已经成为人们日常生活中不可分割的一部分。Android<sup>[1-2]</sup>作为其中的三大主流系统平台之一, 正在被众多的厂商、用户所熟知, 而且由于其众多的用户与自身的开源性, 它也逐渐被各类不法分子与黑客盯上作为攻击的目标。

根据网秦发布<sup>[3]</sup>的 2013 年上半年全球手机安全报告显示, 2013 年上半年新增的手机恶意代码数就为 51 084 款, 感染机器数超过两千万台, 其中 Android 占

了 95%。而中国大陆占其中 31.71% 为世界上受灾最广泛的地区。其原因一部分是因为大陆用户较多, 而更多的应该在于大陆各类 Android 平台繁多, 许多小平台的检查很容易就蒙混过去, 甚至根本就没有检查, 导致各类恶意软件能够迅速传播。对于 Android 平台的架构安全<sup>[4-7]</sup>已经有过很多文章进行过分析, 但具体技术<sup>[8]</sup>却很少有人分析。文中以一种简单的 Android 木马为例, 分析了现在影响最广的信息获取与远程控制两种木马功能的具体实现, 弥补实践上的空白, 期望能够有一种针对性的防范措施。

收稿日期: 2013-12-05

修回日期: 2014-03-12

网络出版时间: 2014-07-28

基金项目: 国家科技支撑计划项目(2011BAK05B03); 信息安全公安部重点实验室开放课题(C12609)

作者简介: 吴玮斌(1991-), 男, 硕士研究生, 研究方向为自然语言处理、信息安全。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140728.1226.035.html>

1 木马总体架构

整个木马分为服务器与客户端两块,服务器用于发布命令以及接收回传的数据。客户端则植入目标手机获取数据。客户端分为四大模块:命令获取与解析,信息获取,数据回传,远程控制<sup>[9]</sup>。功能结构图如图 1 所示。

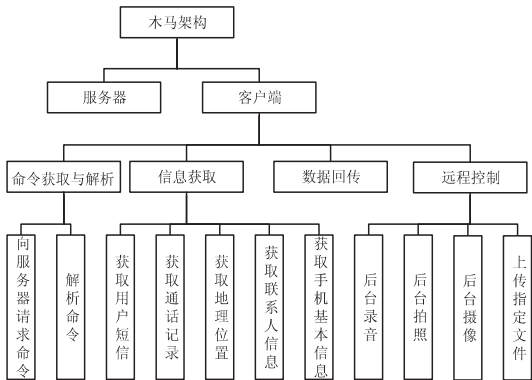


图 1 木马功能结构图

其工作流程为命令获取与解析模块定时向服务器请求控制者设置的命令并解析,然后由信息获取与远程控制模块完成相应的工作,最后由数据回传模块将数据传回服务器保存。如果获取信息时无法连接网络,那么就先存入本地数据库等待用户连接网络后读取并回传数据。具体工作流程如图 2 所示。

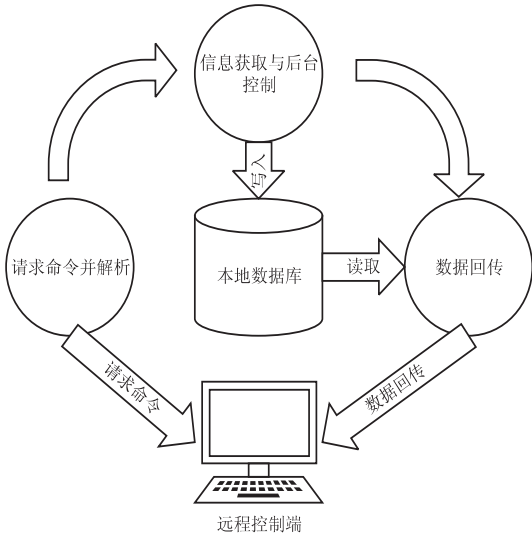


图 2 模块工作流程图

2 实现原理

2.1 命令请求

传统的智能手机木马一般是通过使用短信来达到控制的目的,而其使用的手法大多是将自身的优先级设得非常高而后拦截广播。但是随着 Android 平台上各类安全软件的兴起,这种做法是十分不安全的,控制短信有极大的可能性会被安全软件所拦截。所以文中设计的木马获得命令的方式是通过 HTTP 来进行命令

请求,在服务器端允许客户端两项不同的操作,分别是 upload 与 update。以更新数据与请求命令的 update 为例,其设计如表 1 所示。

表 1 update 操作

参数名称	参数类型	描述
action	String	必填,固定值"update",指明为 update action
imei	String	必填,设备 IMEI 或其他唯一识别码
update_info	JSON String	可选,更新手机基本信息
update_contact	JSON String	可选,更新通讯录
update_sms	JSON String	可选,更新短信内容
update_location	JSON String	可选,更新位置信息
update_command	JSON String	可选,更新命令返回结果
update_call_history	JSON String	可选,更新通话记录

命令请求的实现就是仅提交 action 与 imei 而所有的 update\_\* 都不提交。这样就能向服务器表明该受控端现在在线,然后服务器就能将预先设置好的命令传回客户端执行。

以 10 秒请求间隔为例,具体实现需要定义一个 Handler 然后使用 postDelayed 方法:

```
Handler mHandler=new Handler();
mHandler.postDelayed(mRequestRunnable,10*1000);
```

其中,mRequestRunnable 是自定义的一个 Runnable 类,发送一个包含 action 与 imei 的 HttpRequest 并将返回值解析成具体命令。

2.2 信息获取

根据需求的不同,一个木马所需要获取的数据也不同。各种信息的获取大致可分为六类不同的获取方式:

(1)使用系统类获得:如手机系统版本号、手机型号等基本信息就可以通过 android.os.Build 类获得:

```
String version=android.os.Build.VERSION.RELEASE;
String model=android.os.Build.MODEL;
```

(2)调用 getSystemService 返回相应信息类:如 Wi-Fi 信息可以通过将参数设置为 WIFI\_SERVICE 来获得对应的 WifiManager 类:

```
WifiManager wifiManager=(WifiManager) getSystemService(WIFI_SERVICE);
WifiInfo wifiInfo=wifiManager.getConnectionInfo();
String mWifiInfo=wifiInfo.toString();
```

除了直接返回总体信息以外还能通过对应的 API 返回更具体的信息,如遍历所有能够搜索到的 Wi-Fi 信号:

```
List<ScanResult> scanResults=wifiManager.getScanResults();
for(ScanResult scanResult:scanResults){
    //获得具体信息
}
```

(3)通过监听广播获得<sup>[10]</sup>:如电池电量信息可以通过使用 ACTION\_BATTERY\_CHANGED 这个 Filter

来获取系统广播:

```
BroadcastReceiver batteryLevelReceiver = new BroadcastReceiver() {  
    public void onReceive(Context context, Intent intent) {  
        //通过 intent 中的 level 和 scale 两个 int 类型的附加参数可以  
        //得到对应的电量  
    }  
};  
IntentFilter batteryFilter = new IntentFilter ( Intent. ACTION _  
BATTERY_CHANGED);  
registerReceiver( batteryReceiver, batteryLevelFilter );
```

(4) 通过对应信息查询: 比如 GPS<sup>[11-12]</sup> 更隐蔽的定位手法 Wi-Fi 定位与基站定位可以通过 Wi-Fi 信息或基站信息连网使用公开 API 查询对应的地理位置信息。可以同时使用三种定位方式使得程序可以在各种情况下都能获得地理位置信息。如使用百度的公开 API:

直接使用 `mLocationClient.requestLocation()` 进行请求, 然后重写回传函数 `onReceiveLocation()` 以获取需要的信息即可。

(5) 使用 Content Provider 查询对应信息: 如短信记录就能够使用 `content://sms` 这个 `ContentUri` 通过 `ContentResolver` 类的 `query` 方法查到:

```
Cursor cursor = getContentResolver(). query ( SMS_CONTENT _  
URI, new String[] { " _id", " address", " date", " type", " body", " _  
_id>?", new String[] { String.valueOf( lastId ), " _id DESC" } );
```

这样就能获得一个包含从 `lastId` 开始的所有短信的 ID, 通信号码, 日期, 发送类别 (接收/发送) 以及内容的 `cursor`, 然后对 `cursor` 进行操作就能获得需要的短信内容, 并且实现增量上传。

(6) 文件系统信息<sup>[13]</sup>: 可以使用 `File` 类的 `listFiles` 方法列出文件目录:

```
File folder = new File( fileDir );  
File [ ] sonFiles = folder. listFiles( );
```

这个例子中, 只要遍历 `sonFiles` 就可以获得 `fileDir` 目录下所有文件与文件夹的信息, 实际使用时 `fileDir` 是通过命令传输, 然后返回该目录下所有文件与文件夹信息, 然后控制者能够继续浏览某个文件夹或者命令上传某个文件, 以实现对用户数据的获取。

## 2.3 数据回传

数据回传<sup>[14]</sup> 是通过 HTTP/HTTPs 协议的 POST 请求执行的, 客户端将获取的信息 (包括短信, 联系人, 通话记录等) 按照某种格式组装成 json 形式的数据回传到服务器, 该 json 格式数据包括命令 ID, 所传信息类别, 手机的唯一识别码 (imei) 等内容。如果包含远程控制的结果 (录音, 照片, 录像等) 内容就通过上传文件的方式将其回传, 然后将受控机上的对应文件删

除。

与回传信息的时候不同, 在回传文件的时候要用到 `MultipartEntity`, 然后通过 `FileBody` 来添加文件附件。

```
FileBody file = new FileBody( new File( mFileDir ) );  
reqEntity. addPart( "upload", file );
```

## 2.4 远程控制

虽然现在 2G/3G/Wi-Fi 网络的覆盖率已经相当高, 大部分用户的智能手机都是长时间连入互联网的, 但是也会有不能连入网络的情况存在, 所以文中的木马可以接受提前设定运行时间的远程控制命令, 然后通过定时广播来启动相对应的后台控制 Service。

定时广播主要是通过 `AlarmManager` 实现, 将其设置为 `RTC_WAKEUP` 模式后, 再设置对应的时间与 `PendingIntent` 就可以在指定时间执行后者中的 `Intent` 开启后台服务。

```
PendingIntent pi = PendingIntent. getBroadcast ( getApplication -  
Context(), 0, intent, 0 );
```

```
AlarmManager am = ( AlarmManager ) getSystemService ( A -  
LARM_SERVICE ); am. set ( AlarmManager. RTC_WAKEUP, cal.  
getTimeInMillis(), pi );
```

其中 `cal` 是一个 `Calendar` 类实例, 用于设置命令中的启动时间。不过由于广播在关机或重启后不会保存, 所以为了保证命令能够执行, 还需要使用数据库等手段来将未执行命令保存, 等待重新开机后继续设定广播。文中设计使用 `SharePreference` 保存还未执行的命令, 并在开机时检查这些命令, 将其重新设定或直接执行。

另外由于拍照和摄像在 Android4.0 以后已经无法在后台中运行, 必须在 `Activity` 中有预览界面, 虽然通过设置预览界面的大小可以实现将预览隐藏, 但无法改变其 `Activity` 的本质。于是为了隐藏就需要在用户暗屏时录像拍照, 当屏幕亮起的一瞬间将摄像头 `Activity` 关闭:

首先通过 `PowerManager` 类中的 `isScreenOn()` 方法判断屏幕是否关闭, 如果关着则进行下一步拍照或录像, 并注册一个 `Receiver` 监听屏幕启动的广播:

```
IntentFilter scrStatusIF = new IntentFilter();  
scrStatusIF. addAction( Intent. ACTION_SCREEN_ON );  
registerReceiver( mScreenReceiver, scrStatusIF );
```

其中的 `mScreenReceiver` 是自定义的一个 `BroadcastReceiver` 类, 重写其中 `onReceive` 函数, 判断是否开启摄像头活动, 如果开启着就关闭该活动:

```
if ( SPHelper. getIsCameraOn( context ) ) {  
    CameraActivity. instance. finish();  
}
```

`CameraActivity` 中的 `instance` 是一个其自身的 `this` 指针, 专门用于外部活动或服务将其关闭。

3 测试与分析

测试结果如表 2 所示,其中√表示通过,×表示该功能中有不能使用的部分。项 1 为命令请求,项 2 为信息获取,项 3 为数据回传,项 4 为远程控制。

表 2 测试结果

测试机型	版本号	项 1	项 2	项 3	项 4
SCH-N719	4.1.3	√	√	√	√
HTC Desire HD	2.3.5	√	√	√	√
华为 U8160	2.2.3	√	√	√	√
某山寨机品牌	4.0.3	√	√	√	√

从测试结果可以看出该木马的运行情况良好,各大主流厂商的机器均能够良好运行。测试过程中在一台 ROM 给大幅度修改的山寨机上获取短信时因为短信数据库的格式发生了较大幅度改变导致出现过问题,在对木马进行了一个重新适配后能够正常运行。所以可能在一些大幅度修改的定制 ROM 上使用会出现不可预知的错误。同时从使用效果上来看当受控端在线的时候将请求间隔设置为 10 s 左右就近似可以看成是实时控制了,与短信控制相差无几,甚至在一些情况下能够比短信更快。而且比短信控制更隐蔽不易被发现。

4 结束语

随着移动设备技术的不断发展,其已经在人们的生活中占据了一个极其重要的位置。而 Android 作为其中唯一的一款开源主流操作系统,针对其的攻击显然会非常多。而这些攻击中又以隐私获取与远程控制的危害最大,如何应对这些攻击就成为了当下研究的一个重点。希望文中所提出的木马原型能够对研究起到一定的积极作用。

参考文献:

[1] Developers A. What is android? [J/OL]. 2011. <http://developer.android.com/guide/basics/what-is-android.html>.

[2] 刘仙艳. 移动终端开放平台—Android[J]. 信息通信技术, 2011,5(4):50-53.

[3] 网秦安全公司. 2013 上半年网秦全球手机安全报告[R]. 北京:网秦安全公司,2013.

[4] Enck W,Ongtang M, McDaniel P. Understanding android security[J]. IEEE Security & Privacy,2009,7(1):50-57.

[5] Enck W,Octeau D,McDaniel P,et al. A study of android application security[C]//Proceedings of the 20th USENIX conference on security. [s. l. ]:[s. n. ],2011:21-44.

[6] Ongtang M,McLaughlin S E,Enck W,et al. Semantically rich application-centric security in Android[J]. Security and Communication Networks,2011,5(6):658-673.

[7] 宋 杰,党李成,郭振朝,等. Android OS 手机平台的安全机制分析和应用研究[J]. 计算机技术与发展,2010,20(6):152-155.

[8] 董 蕾,黄淑华,尹浩然,等. 基于 Android 平台的手机木马关键技术分析[J]. 信息网络安全,2012(11):63-65.

[9] 耿东久,索 岳,陈 渝,等. 基于 Android 手机的远程访问和控制系统[J]. 计算机应用,2011,31(2):559-561.

[10] 蔡罗成. Android 后台监听实现机制浅析[J]. 信息安全与通信保密,2010(6):39-41.

[11] 赵建勋. 基于 Android 平台的移动位置服务的开发与实现[J]. 现代商贸工业,2010,22(20):271-272.

[12] 黄志勇,赵 雯. 基于 Android 平台的移动位置信息服务开发研究[J]. 自动化技术与应用,2011,30(12):22-26.

[13] 温 敏,艾丽蓉,王志国. Android 智能手机系统中文件实时监控的研究与实现[J]. 科学技术与工程,2009,9(7):1716-1719.

[14] 黄锦川,金炜东. 基于 Android 平台 Web 服务的应用研究[J]. 铁路计算机应用,2010,19(11):24-27.

(上接第 146 页)

[5] Flanagan J L, Johnston J, Zahn R. Computer-steered microphone arrays for sound transduction in large rooms[J]. Journal of Acoustical Society of American,1985,78(5):1508-1518.

[6] Griffiths L J, Jim C W. An alternative approach to linearly constrained adaptive beamforming[J]. IEEE Transactions on Antennas and Propagation,1982,30(1):27-34.

[7] Zelinski R. A microphone array with adaptive post-filtering for noise reduction in reverberant rooms[C]//Proc of international conference on acoustics,speech and signal processing. New York:IEEE,1988:2578-2581.

[8] 马晓红,殷福亮,陆晓燕,等. 基于小波变换的传声器阵列语音增强方法[J]. 大连理工大学学报,2003,43(4):511-515.

[9] 王冬霞,殷福亮. 联合波束形成与谱减法的麦克风阵列语

音增强算法[J]. 大连理工大学学报,2006,46(1):121-126.

[10] 张宝琳,张玲华,林志敏. 基于多分辨率小波阈值去噪的响度补偿方法[J]. 计算机技术与发展,2012,22(12):83-86.

[11] 徐耀华,王 刚,郭 英. 基于时频阈值的小波包语音增强算法[J]. 电子与信息学报,2008,30(6):1363-1366.

[12] 孙延奎. 小波分析及其应用[M]. 北京:机械工业出版社,2005.

[13] 李海东,李 青. 基于阈值法的小波去噪算法研究[J]. 计算机技术与发展,2009,19(7):56-58.

[14] 李如玮,鲍长春,窦慧晶. 基于双正交小波包分解的自适应阈值语音增强[J]. 仪器仪表学报,2008,29(10):2135-2140.

# 基于Android平台的手机木马的设计与实现

作者:

吴玮斌, 孟魁, 徐林, 刘功申, WU Wei-bin, MENG Kui, XU Lin, LIU Gong-shen

作者单位:

吴玮斌, 孟魁, 刘功申, WU Wei-bin, MENG Kui, LIU Gong-shen(上海交通大学 电子信息与电气工程学院, 上海, 200240), 徐林, XU Lin(公安部第三研究所信息安全公安部重点实验室, 上海, 200120)

刊名:

计算机技术与发展 

英文刊名:

Computer Technology and Development

年, 卷(期):

2014(10)

引用本文格式: 吴玮斌. 孟魁. 徐林. 刘功申. WU Wei-bin. MENG Kui. XU Lin. LIU Gong-shen 基于Android平台的手机木马的设计与实现[期刊论文]-计算机技术与发展 2014(10)