doi:10.3969/j.issn.1001-0505.2013.06.006

一种 Android 平台恶意软件静态检测方法

秦中元[1,2] 徐毓青[1] 梁 彪[3] 张群芳[4] 黄 杰[1]

([1] 东南大学信息安全研究中心,南京 210096)
([2] 信息网络安全公安部重点实验室,上海 201204)
([3] 南京三宝科技股份有限公司,南京 210049)
([4] 南京炮兵学院计算机教研室,南京 211132)

摘要:为了有效地检测 Android 平台的恶意软件,提出一种基于危险权限和行为分析的静态综合检测方法.对已检测过的应用程序包(APK),提取消息摘要的 MD5 值作为签名用来进行快速匹配和判定;未检测过的 APK 根据权限和行为分析来判定,首先通过检测是否申请危险权限进行预判,然后进行污点传播和语义分析,以检测出 APK 中是否存在隐私窃取和恶意扣费行为.与杀毒软件只能检测出已知的恶意软件不同,本系统不依赖于病毒库的收集和更新,可对已知恶意软件变种和未知恶意软件进行有效检测.实验中成功检测出了隐私窃取与恶意扣费的恶意行为,证明了本系统的有效性.

关键词:安卓;恶意软件;静态检测;权限;行为分析
中图分类号:TP309 **文献标志码**:A **文章编号**:1001-0505(2013)06-1162-06

An Android malware static detection method

Qin Zhongyuan[1,2] Xu Yuqing[1] Liang Biao[3] Zhang Qunfang[4] Huang Jie[1]

([1]Information Security Research Center, Southeast University, Nanjing 210096, China)
([2]Key Laboratory of Information Network Security of Ministry of Public Security, Shanghai 201204, China)
([3]Nanjing Sample Technology Co., Ltd, Nanjing 210049, China)
([4]Computer Department, Nanjing Institute of Artillery Corps, Nanjing 211132, China)

Abstract: In order to efficiently detect malicious software on Android, an integrated static detection method is proposed based on dangerous permissions and behavior analyses. For the application package (APK) which has been detected before, its MD5 value is extracted as the signature for fast match and decision. For those which have not been detected, permission and behavior analyses are used to detect whether it is malware or not. First, a pre-decision is made according to whether dangerous permissions are applied. Secondly, taint propagation and semantic analyses are conducted to detect the behavior of stealing private information and financial over-charge in APK. The proposed system does not depend on the collection and update of the virus database and can efficiently detect the variants of known and unknown malware, which is different from the anti-virus software that can only detect known malware. The experimental results show that malwares with privacy stealing and malicious extra charges are successfully detected, which proves the effectiveness of the system.

Key words: Android; malware; static detection; permission; behavior analysis

智能手机是移动互联网的重要载体,随着移动互联网的迅速发展,智能手机市场规模也不断增长.Android 系统由于其开源性,任何组织或个人编写的软件都可以上传到“应用商店”中,供用户

收稿日期:2013-04-22. **作者简介**:秦中元(1974—),男,博士,副教授,zyqin@seu.edu.cn.
基金项目:国家高技术研究发展计划(863 计划)资助项目(2013AA014001)、国家发改委信息安全专项资助项目、信息网络安全公安部重点实验室开放课题资助项目(C13611).
引文格式:秦中元,徐毓青,梁彪,等.一种 Android 平台恶意软件静态检测方法[J].东南大学学报:自然科学版,2013,43(6):1162-1167.[doi:10.3969/j.issn.1001-0505.2013.06.006]

下载并安装使用,因此已经占据全球出货量 75% 的智能手机[1]. 但是,智能手机市场的迅速发展也带来了与用户密切相关的恶意软件问题.

目前国际上对于智能手机恶意软件检测的研究所使用的方法主要可分为 2 类:基于规则的方法和基于异常的检测方法.

基于规则的方法将权限组合、行为足迹、作者信息等作为判定的规则. 例如 Enck 等[2]定义了各种潜在的危险权限组合作为规则,通过检查特定的危险权限组合来中止潜在不安全应用的安装. Zhou 等[3]提出了一个基于权限的行为足迹匹配来检测 Android 恶意软件已知家族的样本,并应用启发式过滤来检测未知的 Android 恶意软件. Zhou 等[4]使用模糊哈希来检测第三方 Android 应用商店中的重打包的应用程序(潜在的恶意软件),并利用指令和程序作者信息来进行检测.

基于异常的检测方法选取了权限、系统调用和功耗等作为学习的特征,并利用聚类和分类算法来区分正常和恶意的应用. 例如 Schmidt 等[5]实现了 Android 异常检测系统,该系统从 Android 系统的各个层次获取系统数据作为检测依据,并使用 Ad-Hoc 网络中的联合异常检测算法对数据进行处理. Shabtai 等[6-7]提出了基于行为的 Android 恶意软件检测系统 Andromaly,测试了 CPU 消耗、通过 Wi-Fi 发送的数据包数目、正在运行的进程数目、按键以及应用程序启动等特征来寻找最典型的特征集合,并应用了一些不同的机器学习算法,如 Logistic 回归分析(logistic regression)和贝叶斯网络(Bayesian networks)来对应用程序分类. Zhao 等[8]提出了基于行为的恶意软件检测框架 AntiMalDroid,使用支持向量机算法,动态地记录行为序列作为特征.

这些检测方法涉及了静态分析和动态分析. 静态分析利用程序的静态语法或结构属性来判定其恶意性,而动态分析主要通过监视程序的执行来检测恶意行为. 静态分析与动态分析相比,具有以下优点[5]:

1) 静态分析允许进行全面分析. 静态分析不受一个程序的特定执行过程约束,并且适用于程序的所有执行过程. 相反地,动态分析技术只允许对符合选定的测试案例的行为进行检查.

2) 结果在执行过程之前被给出,恶意行为难以伪装.

3) 没有运行时负担. 静态分析方法能够减少成本并且提高性能. 尽管使用动态分析能够提供恶意软件的综合观察,但检查中容易受到环境部署的影响,人工成本也较高.

基于此,本文对 Android 平台上恶意软件检测进行了研究,提出一种静态的综合检测方法. 对于已检测过的 APK 文件,提取 MD5 值作为签名,在处理过程中利用 MD5 值进行快速匹配和判定. 对于未检测过的 APK 文件,根据对权限和恶意行为总结的规则来进行分析和判定. 实验中成功地从恶意软件中检测出了隐私窃取与恶意扣费行为,证明了本系统的有效性.

1　系统设计

1.1　系统框架

本文系统总体结构如图 1 所示. 该系统主要包括 5 个模块,分别为:

1) 已分析程序判断模块. 该模块计算被检测 APK 包的 MD5 值,并与已分析程序库中存储的 MD5 值进行比较,如果该 MD5 值已存在,表明该 APK 包已检测过,则跳过检测分析过程,直接进入结果输出模块.

2) 预处理模块. 该模块实现解压和反编译.

3) 权限分析模块. 对解压和反编译处理后得到的 AndroidManifest 文件根据是否存在危险权限进行分析,如果不存在则直接进入结果输出模块.

4) 行为分析模块. 根据恶意行为规则库中定义的恶意行为进行分析.

5) 结果处理模块. 将已分析程序检测结果存入已分析程序库,如果被检测文件具有恶意行为,则提示用户进行删除或隔离操作.

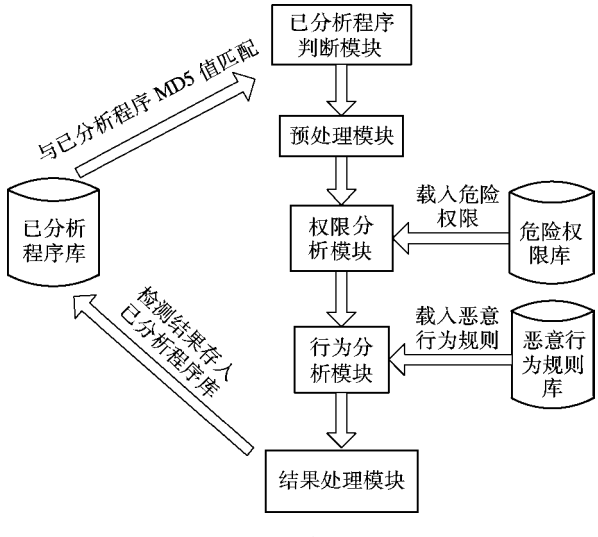

图 1　系统模块图

1.2　系统处理流程

系统具体处理流程为:

① 用户计算待检测 APK 文件的 MD5 值,并与已分析程序库比较. 如果该 MD5 值在已分析程序库中存在,表明已检测过,处理结束.

② 将 APK 包解压,得到 AndroidManifest. xml 文件和 classes. dex 文件.

③ 使用 AXMLPrinter2,dex2jar,jd-gui 等工具对 AndroidManifest. xml 文件进行反编译,并将 classes. dex 文件转换成 Java 文件.

④ 对反编译后的 AndroidManifest. xml 文件进行分析. 如果不存在危险权限,处理结束.

⑤ 对反编译得到的 Java 文件进行行为检测. 如果存在恶意行为,则提示用户选择处理操作,执行删除或隔离文件的操作.

⑥ 将检测结果存入已分析程序库,处理结束.

2 恶意软件检测

2.1 权限分析

2.1.1 Android 权限机制

Android 的权限机制限制了一个应用程序能够执行的特定操作. Android 有 100 多个内置的权限,这些权限控制从拨打电话、拍照、使用网络到监听按键,甚至是永久禁用手机的操作. 为了获取权限,应用程序必须明确地申请这个权限. Android 应用程序需要在 AndroidManifest. xml 中使用〈uses-permission〉标签来指定所申请的权限.

2.1.2 危险权限

由于每个 Android 应用程序都在 AndroidManifest. xml 文件中明确地申请了其所需要的权限,因此可以利用实现恶意软件功能所需要的一些危险权限来排除安全的应用程序.

系统所考虑的危险权限包括读取隐私信息、收发短信、拨打电话、网络设置等. 限于篇幅,表 1 列出了与获取隐私信息、收发短信和拨打电话相关的危险权限.

表 1 危险权限

危险权限类别	权限列表
获取隐私信息	ACCESS_COARSE_LOCATION ACCESS_FINE_LOCATION READ_CONTACTS READ_OWNER_DATA READ_PHONE_STATE READ_SMS
收发短信	RECEIVE_SMS SEND_SMS
拨打电话	CALL_PHONE PROCESS_OUTGOING_CALLS

以获取隐私信息和收发短信的危险权限为例,说明对这些危险权限的选取和划分依据.

1) ACCESS_COARSE_LOCATION 权限允许应用程序访问大概的位置.

2) ACCESS_FINE_LOCATION 权限用于访问精确的位置源.

3) READ_CONTACTS 权限允许应用程序读取用户手机上存储的所有联系人数据.

4) READ_OWNER_DATA 权限允许应用程序读取用户手机上存储的手机所有者数据.

5) READ_PHONE_STATE 权限允许应用程序访问设备的手机功能.

6) READ_SMS 权限允许恶意软件读取用户手机或 SIM 卡中存储的短信.

收发短信危险权限经常被 Android 恶意软件利用. 一种典型的恶意软件是在用户不知情的情况下向固定的增值服务号码发送订制扣费业务短信,给用户带来额外的费用. 为了隐藏这一行为,恶意软件监视接收到的短信,并且将话费账单的通知短信自动删除. 因此恶意软件需要分别申请 SEND_SMS 权限和 RECEIVE_SMS 权限.

权限分析根据选取的危险权限对反编译后的 AndroidManifest. xml 文件进行扫描,不存在这些危险权限申请的应用程序将无需执行后续的行为分析,被判定为正常.

2.2 行为分析

Android 应用程序的主要代码包括在 classes. dex 文件中,该文件是源码编译后生成的 Java 字节码. 与 AndroidManifest. xml 文件相似,本系统对其进行反编译后再作分析.

中国互联网协会反网络病毒联盟发布的《移动互联网恶意代码描述规范》[9] 中指出,移动互联网恶意代码往往被用于窃取用户个人隐私信息,非法订购各类增值业务,造成用户直接经济损失. 本文系统的行为分析模块主要针对隐私窃取和恶意扣费来进行分析.

隐私窃取行为窃取的信息包括手机相关信息、地理位置信息等,将隐私信息发送出去的方式主要有短信和网络;恶意扣费行为主要包括自动发送订制收费业务短信和拦截服务号码发送的确认、账单相关短信. 针对这些恶意行为,本系统采用污点传播分析和语义分析方法.

2.2.1 污点传播分析

污点传播分析方法适用于检测程序是否具有隐私窃取行为. 本文考虑的隐私窃取行为的信息对象有 2 种类别. 一种是与手机相关的信息,包括本

机电话号码、IMEI 号、IMSI 号和 ICC-ID. 这些内容的获取分别通过 TelephonyManager 类的 getLine1Number( ) 方法、getDeviceID( ) 方法、getSubscriberId( ) 方法和 getSimSerialNumber( ) 方法来实现. 因此,对于隐私窃取的恶意行为,将上述方法选取为源函数. 隐私窃取恶意行为的另一种信息对象是地理位置信息,包括经度、纬度、基站编号和本地区域代码(考虑接入网为 GSM),其相应的源函数选取为 Location 类的 getLatitude( ) 方法和 getLongitude( ) 方法以及 GsmCellLocation 类的 getCid( ) 和 getLac( ) 方法. Android 与外界交互主要有短信、网络等途径. 因此,在选取锚点函数(sink 函数)时,将与这些交互方式相关的函数设置为锚点函数. 对于短信发送隐私信息的方式,选取 SmsManager 类的发送短信函数 sendTextMessage( )、sendDataMessage( ) 和 sendMultipartMessage( ) 作为锚点函数. 对于通过网络发送隐私数据的方式,由于 Android 对于 HTTP 网络通信提供了 httpURLConnection 接口和 httpclient 接口,选取的函数有 DataOutputStream 类的 writeBytes( ) 和 HttpClient 的 execute( ) 等函数.

对源函数到锚点函数的数据流向的分析判断,依赖于对程序结构的分析. 源节点到目标节点之间的路径可以通过遍历控制流图的基本块得到. 路径生成与污点传播算法采用孔德光等[10] 提出的算法.

路径生成算法的输入为源节点 src(源函数)与目标节点 tgt(锚点函数),输出为 src 到 tgt 的路径. 首先识别源节点和目标节点所在的基本块;如果在同一个基本块内,则直接在基本块内寻找从源节点到目标节点的路径;如果源节点和目标节点在同一个函数但不在同一个基本块中,则寻找该函数内基本块之间的路径;如果源节点和目标节点在不同的函数体中,则迭代展开函数调用,寻找不同函数调用间的路径.

污点传播过程用于跟踪污染和未污染数据的变化过程. 算法的输入为当前语句 stmt,输出为标记该条语句中从源数据(source)到目的数据(target)的污点传播过程. 其标记过程如图 2 所示,根据当前语句的类型来判定目的数据的污染状态.

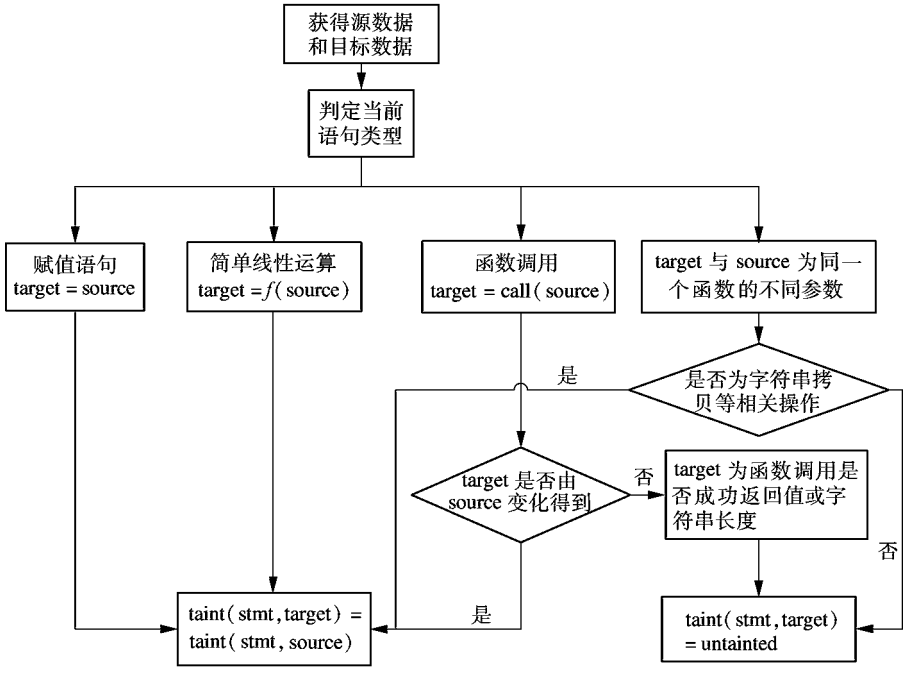

图 2　污点传播算法

2.2.2　语义分析

根据对恶意扣费应用程序的分析发现,其恶意行为的主要表现有:

1) 申请了收发短信权限.

2) 向特定号码发送短信.

3) 注册监听消息,并屏蔽来自特殊号码的短信.

因此,针对恶意扣费行为,采用基于特殊字符串和函数的语义分析来进行检查.

具体地,针对恶意发送订制服务短信同时拦截运营商回复短信的行为,对其进行基于语义分析的检测,算法描述如下:

① 检查 AndroidManifest 文件中是否申请了 android. permission. RECEIVE_SMS 接收短信权限和 android. permission. SEND_SMS 发送短信权限.

② 在程序中对发送短信的 API 函数调用进行定位,这些函数包括 sendTextMessage( ), sendDataMessage( )和 sendMultipartMessage( ).

③ 对关键函数调用的第 1 个参数(发送信息的目的号码)进行匹配检查,看是否为 106 开头的数字.

④ 检查是否有注册监听 android. provider. Telephony. SMS_RECEIVED 事件.

⑤ 在④中相应类的 abortBroadcast( )函数的上下文中对“10086”,“10000”,“10010”和 106 开头的特定号码进行匹配检查.

3 实验结果

本文在处理器为 AMD Phenom(tm) Ⅱ X4 970,主频为 3.5 GHz,RAM 为 4 GB,操作系统为 Windows 7(32 位)的计算机上进行了实验. 选取了 10 个恶意软件样本(编号 1 ~ 10),利用本文的检测系统对这些应用程序进行分析.

利用本文系统对每个样本的 AndroidManifest 文件进行分析,得出每个样本所申请的危险权限类别,如表 2 所示.

表 2 权限分析结果

编号	MD5 值	危险权限			
		网络	短信	隐私获取	自启动
1	944cbd53a174dcbb0e1e3d0832cac465	✓	✓	✓	
2	6a17bfaa44d177a681b725909ee55247	✓		✓	✓
3	badd63bce3ab7e622abb088b29f15fd4	✓	✓	✓	
4	c0e6ba0e1b757e3c506a02282ffc5b41	✓	✓	✓	
5	1d33631145ab19b51b2e404845d3256c	✓	✓	✓	
6	10629b68370e78def7f3f93ab95d2c52	✓	✓	✓	
7	c8d2c674621a8694fdf26306dd08ded9	✓		✓	✓
8	992e682eeb9733d26645060c98e81feb	✓		✓	✓
9	16c69031a4891c34f0458402de268975	✓	✓	✓	
10	8f552acc0bc8e65852a22c2cdf5f4de8	✓		✓	✓

由表 2 可看出:

1) 样本 1,3,4,5,6,9 都申请了与短信相关的权限,而且其中除了样本 3 以外都同时申请了接收短信和发送短信的权限,行为分析也进一步判定样本 1,4,5,9 具有恶意扣费行为,表明收、发短信的权限与恶意扣费行为实现的相关性.

2) 所有样本都申请了与读取用户隐私相关的权限和与网络相关的权限,表明这 2 种权限是恶意软件通常需要利用的.

3) 样本 2,7,8,10 申请了与自启动相关的权限.

对样本检测出的恶意行为进行统计分析,如表 3 所示.

表 3 行为分析结果

编号	MD5 值	恶意扣费	隐私窃取	
			手机信息	地理位置
1	944cbd53a174dcbb0e1e3d0832cac465	✓		
2	6a17bfaa44d177a681b725909ee55247		✓	
3	badd63bce3ab7e622abb088b29f15fd4		✓	
4	c0e6ba0e1b757e3c506a02282ffc5b41	✓		
5	1d33631145ab19b51b2e404845d3256c	✓		
6	10629b68370e78def7f3f93ab95d2c52		✓	✓
7	c8d2c674621a8694fdf26306dd08ded9		✓	
8	992e682eeb9733d26645060c98e81feb		✓	✓
9	16c69031a4891c34f0458402de268975	✓		
10	8f552acc0bc8e65852a22c2cdf5f4de8		✓	

由表 3 可看出,在所检测的 10 个恶意样本中,样本 1,4,5,9 具有恶意扣费行为. 这个结果也与权限申请情况符合,因为这 4 个样本都同时申请了接收短信和发送短信的权限.

样本 2,3,6,7,8,10 检测出具有隐私窃取的恶意行为,这 6 个样本都窃取了与手机信息相关的隐私,样本 6 和 8 同时还窃取了地理位置信息.

下面对检测出的一个样本 zsone. apk 进行具体分析.

1) 基本信息. 名称为 zsone. apk. 其 MD5 值为 944cbd53a174dcbb0e1e3d0832cac465.

2) 该软件申请的危险权限. android. permission. INTERNET, android. permission. ACCESS _ COARSE _ LOCATION, android. permission. RECEIVE_SMS 和 android. permission. SEND_SMS.

3) 恶意行为信息. 检测到恶意扣费行为. 发送的目标号码及内容为:10621900,“M6307AHD”;10626213,“aAHD”;106691819,“95pAHD”;10665123085,“58#28AHD”. 屏蔽的号码为:10086,10000,10010,1066133,10655133.

对其进行反编译得到的代码中的关键代码如下所示:

```
private void sendCM( )
{SmsManager. getDefault( ). sendTextMessage(“10621900”, null,”M6307AHD”,PendingIntent. getBroadcast( this. context,0,new Intext( ),0),null);}
private void sendCM( )
{SmsManager. getDefault( ). sendTextMessage(“10626213”, null,”aAHD”,PendingIntent. getBroadcast( this. context,0, new Intext( ),0),null);}
…}//以上代码实现了向号码 10621900 等发送恶意扣费短信
```

的行为.

```
try
{String str = ocalSmsMessage. getDisplayOriginatingAddress( );
if (("10086". equals(str)) ‖ ("10000". equals(str)) ‖ ("10010". equals(str)) ‖ ("1066133". equals(str)) ‖ ("10655133". equals(str)))
abortBroadcast( );
k + +;}//以上代码实现了拦截来自号码 10086 等的短信的行为.
```

通过分析上面的反编译结果可得出,该样本可以实现向特定号码发送订制扣费服务的短信,并能屏蔽来自特定号码的短信,使用户无法收到服务提供者的确认或通知短信,在用户不知情的情况下造成恶意扣费,与检测结果相同.

4　结语

本文对现有的智能手机平台恶意软件检测工作进行了总结,提出了一种综合检测方法. 对于已检测过的 APK 文件,提取 MD5 值作为签名,在处理过程中利用 MD5 值进行快速匹配和判定. 对于未检测过的 APK 文件,根据对权限和恶意行为总结的规则来进行分析和判定.

本文方法与基于签名的方法相比,具有可以检测出已知恶意软件变种和未知恶意软件的优势. 与基于异常的检测方法相比,更准确地定义了实际的恶意行为,而且避免了训练阶段的特征选择问题.

参考文献(References)

[1] IDC. Android marks fourth anniversary since launch with 75.0% market share in third quarter, according to IDC [EB/OL]. (2012-11-01) [2013-01-02]. http://www. idc. com/getdoc. jsp? containerId = prUS23771812.

[2] Enck W, Ongtang M, McDaniel P. On lightweight mobile phone application certification[C]//*Proceedings of the* 16*th ACM Conference on Computer and Communications Security*. Chicago, IL, USA, 2009:235 - 245.

[3] Zhou Yajin, Wang Zhi, Zhou Wu, et al. Hey, you, get off of my market: detecting malicious apps in official and alternative Android markets[C]//*Proc of the* 19*th Annual Network and Distributed System Security Symposium*. San Diego, CA, USA: 2012:1 - 13.

[4] Zhou Wu, Zhou Yajin, Jiang Xuxian, et al. Detecting repackaged smartphone applications in third-party Android marketplaces [C]//*Proceedings of the Second ACM Conference on Data and Application Security and Privacy*. San Antonio, TX, USA, 2012: 317 - 326.

[5] Schmidt A-D, Bye R, Schmidt H-G, et al. Static analysis of executables for collaborative malware detection on Android[C]//2009 *IEEE International Conference on Communications*. Dresden, Germany, 2009:1 - 5.

[6] Shabtai A, Elovici Y. Applying behavioral detection on Android-based devices[C]//3*rd International Conference on Mobile Wireless Middleware, Operating Systems, and Applications*. Chicago, IL, USA, 2010:235 - 249.

[7] Shabtai A, Kanonov U, Elovici Y, et al. "Andromaly": a behavioral malware detection framework for android devices[J]. *Journal of Intelligent Information Systems*, 2012, **38**(1):161 - 190.

[8] Zhao Min, Ge Fangbin, Zhang Tao, et al. AntiMalDroid: an efficient SVM-based malware detection framework for android[C]//2*nd International Conference on Information Computing and Applications*. Qinhuangdao, China, 2011:158 - 166.

[9] 工业和信息化部. 移动互联网恶意代码描述规范[EB/OL]. (2011-05-14)[2013-01-02]. http://wenku. baidu. com/view/2978e18ccc22bcd126ff0c90. html.

[10] 孔德光,郑烇,帅建梅,等. 基于污点分析的源代码脆弱性检测技术[J]. 小型微型计算机系统,2009,30(1):78 - 82.
Kong Deguang, Zheng Quan, Shuai Jianmei, et al. Source code vulnerability detection technology based on taint analysis[J]. *Journal of Chinese Computer Systems*, 2009, **30**(1):78 - 82. (in Chinese)