doi :10.3969/j.issn.1671-1122.2014.07.006

基于权限分析的 Android 应用程序检测系统

张金鑫，杨晓辉
（东南大学信息安全研究中心，江苏南京 210096）

摘 要：Android 系统在应用程序安装时仅给予粗略的权限提示界面，此界面不仅权限条目不全，而且解释异常粗略，普通用户完全看不懂，但基于使用需要，只能盲目确定授权。市面上的一些例如手机金山卫士，腾讯手机管家等管理软件，对于应用权限信息的查询要么权限条目远少于实际申请，要么权限解释一样粗略难懂，要么干脆就是直接调用 Android 系统 settings 下的粗略权限列表。

通过研究 Android 的安全机制，在分析了上述现象可能导致的潜在安全隐患的基础上，文章设计开发了一种结合电脑端和手机端，能够对未安装的 APK 文件和已安装的 APP 应用程序进行深入权限检测系统。此系统可以检测出应用软件所申请的精确的权限个数和详细的权限列表，并通过建立数据库的方法给每条权限以及可能引起的安全问题辅以详尽、易懂的说明，使无专业知识的普通用户也可以弄懂所申请权限的作用，提高应用程序使用者的安全意识。此外，此系统还能提供用户针对某条敏感权限进行应用筛选，即列出手机内使用该敏感权限的所有应用，协助用户排查恶意软件，保护系统安全。

针对 Android 平台开放性带来的用户隐私泄露和财产损失的问题，文章通过对 Android 安全机制的分析，给出了一种在电脑端和手机端的基于权限分析的 Android 应用程序检测系统。该系统能检测出各种应用的权限信息，也能检测出具有某条敏感权限的所有应用程序，为用户提供再判断的机会，可以更全面的保障用户信息和财产安全。

关键词：Android；安全机制；权限检测
中图分类号：TP309 **文献标识码：**A **文章编号：**1671-1122（2014）07-0030-05

A Detection System of Android Application based on Permission Analysis

ZHANG Jin-xin, YANG Xiao-hui
(*Research Center of Information Security, Southeast University, Nanjing Jiangsu* 210096,*China*)

Abstract: As the openness of the Android platform lead to the privacy leaks and property damage of users, a novel detection system based on permission analysis for Android applications is proposed in this paper, which can be incorporated with computer terminals and mobile terminals. The proposed detection system can not only detect the whole permission information of applications but also help users check all the applications possess sensitive permission. In addition, the detection system provides secondary judgement so that the information and property security of the users are guaranteed.

Through the security mechanism of Android, based on the potential safety hazard analysis of the phenomenon, this paper designs and developes a system which could dectect the uninstalled APK files and the installed APP application. This system can detect the application software for the accurate access number and detailed list of permissions, and through the method of establishing the database for each authority and supplemented by security problems can be caused in detail, to understand instructions, so that ordinary users without professional knowledge can also understand the right to apply for a role, improve safety awareness application the user of the program. In addition, the system also provides user application screening for sensitive permissions, which lists all the applications that uses the sensitive permissions, so as to assist the user to check the malicious software.

Key words: Android; security mechanism; permission detection

收稿日期：2014-04-22
基金项目：国家发改委信息安全专项
作者简介：张金鑫（1990-），男，江苏，硕士研究生，主要研究方向：智能终端安全；杨晓辉（1968-），女，江苏，副教授，博士，主要研究方向：通信和无线网络安全。

1 概述

1.1 Android安全现状

Android 是由 Google 及其开放手机联盟推出的基于 Linux 系统发展而来的开放源代码的移动操作系统 [1]。目前，Android 已被广泛应用于手机，平板电脑和笔记本等智能移动设备。根据市场研究公司 Canalys 5 月 30 日发布的数据报表，2013 年第一季度全球智能移动设备出货总量为 3.087 亿部，同比增长 37.4%。谷歌的 Android 操作系统在第一季度全球智能移动设备市场上所占份额为 59.5%；苹果 iOS 次之，为 19.3%；微软 Windows 和 Windows Phone 为 18.1%[2]，由此可见 Android 占据移动市场的主导地位。

Android 由于其系统的开源性、市场的开放性，方便个人自由的发布开发的应用程序。一方面为程序开发者提供了发布开发应用的便利，另一方面，由于缺少对发布者有效监管的安全机制，这也为不法分子提供了方便。缺乏安全知识的普通用户极易下载安装这些恶意软件，可能导致如用户个人隐私信息的泄露，莫名的 SP 增值服务费，更为严重的甚至会导致手机无法正常使用等严重后果。据网秦公司《2012 年第三季度全球手机安全报告》显示，目前手机恶意软件主要集中在 Android 平台，感染比例高达 94% 以上。金山手机毒霸 [3] 对 8 万个常用 Android 应用软件申请和使用 Android 系统权限的情况进行了详细分析，近一半软件申请、使用系统权限和该软件的正常功能毫无关联。据金山毒霸安全中心统计，2013 年第一季度，仅 Android 平台，监测到的手机恶意软件就已达到去年全年的 3 倍之多，每天超过 41 万 Android 手机运行了窃取隐私的软件。因此，深入了解并时时掌握应用软件的权限信息对手机用户个人的隐私安全保护尤为重要。

1.2 本文工作

Android 系统在应用程序安装时仅给予粗略的权限提示界面，此界面不仅权限条目不全，而且解释异常粗略，普通用户完全看不懂，但基于使用需要，只能盲目确定授权。市面上的一些例如手机金山卫士，腾讯手机管家等管理软件，对于应用权限信息的查询要么权限条目远少于实际申请，要么权限解释一样粗略难懂，要么干脆就是直接调用 Android 系统 settings 下的粗略权限列表。

通过研究 Android 的安全机制，在分析了上述现象可能导致的潜在安全隐患的基础上，本文设计开发了一种结合电脑端和手机端，能够对未安装的 APK 文件和已安装的 APP 应用程序进行深入权限检测系统。此系统可以检测出应用软件所申请的精确的权限个数和详细的权限列表，并通过建立数据库的方法给每条权限以及可能引起的安全问题辅以详尽、易懂的说明，使无专业知识的普通用户也可以弄懂所申请权限的作用，提高应用程序使用者的安全意识。此外，此系统还能提供用户针对某条敏感权限进行应用筛选，即列出手机内使用该敏感权限的所有应用，协助用户排查恶意软件，保护系统安全。

2 Android 特有的安全机制

Android 系统采用的是 Linux2.6 版本内核，并采用 Dalvik 虚拟机作为应用程序运行环境的可跨越多种平台的操作系统。Android 系统是分层的体系结构 [4]，自底向上包含五层，分别是 Linux 内核、本地库、Android 运行时环境、应用框架和应用程序。Google 在设计开发 Android 操作系统时，除了继承了 Linux 的设计思想，在各层设置了相应的安全防范机制外，还设置了签名和程序权限控制两种 Android 特有的安全机制。

2.1 签名机制

Android 系统的所有应用程序都必须有数字证书，没有数字证书的应用程序，系统不予安装。不同于其他平台，Android 应用程序签名不仅标明了 APK 的发布者，并且提供了程序完整性和可靠性的验证。任何试图篡改 APK 文件的不法分子，都必须对 APK 进行重新签名。而在原作者的签名私钥不泄密的前提下，仿冒签名几乎是不可能与原签名完全一致的，具有唯一性。签名机制在应用程序更新升级中起到保护作用，只有在两者的签名信息完全相同的情况下，系统才会允许进行更新升级操作，否则会禁止此次更新，进一步保护了系统的安全。

2.2 应用程序权限控制机制

权限控制是 Android 系统应用程序安全机制最为核心的机制。Android 通过实施基于权限的安全策略来处理安全问题，即采用权限来限制应用程序安装，并且应用程序只能访问权限内的 API 和资源。Android 定义了 135 种系统权限，分为 4 个保护级别 [5]，分别是 :normal，dangerous，

signature，signatureOrSystem。所有的权限及相关功能说明都可以在 Android 系统的开发说明文档中进行查看 [6]。

在默认情况下，Android 应用程序不具有任何权限。应用程序在运行时涉及的权限需要在 APK 的 AndroidManifest.xml 文件中通过 uses-permission 标签设置权限声明。在安装时，Android 应用程序包管理器会提示用户应用程序权限的申请，只有用户同意授权后，安装才开始进行，用户不同意将会取消安装。在成功安装后运行时，系统会根据固化权限信息对程序访问资源的请求做出答复，如果有对应权限，则访问成功；否则应用将被系统强制关闭，具体过程如图 1 所示。

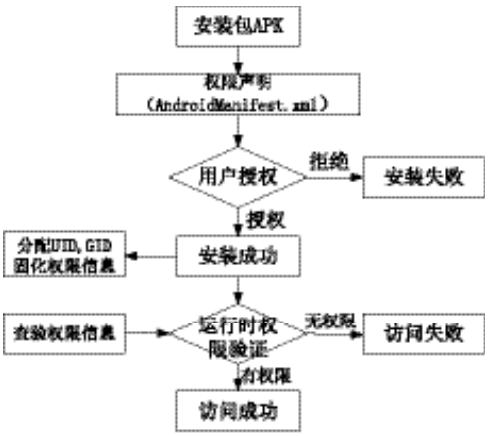

图1 应用安装权限声明与验证过程

Android 权限机制存在明显的安全缺陷。基于权限控制的 Android 系统把安全风险完全推给用户来承担，让用户来掌握权限的授予工作。权限一经被授权给应用程序后，在应用程序的生命期间，即使声明此权限的源程序被删除 [7]，它也不会被移除。这种粗粒度的全授 / 不授，一经授予永久可用的权限机制导致了潜在的安全缺陷。要想降低风险，则要求用户在应用安装授权时具有分辨权限信息的能力，但这对于普通用户来说，无疑是相当困难的。对于此权限机制可能造成的安全隐患，Android 系统在应用程序安装时系统仅仅给予粗略的权限提示界面，如图 2 所示。此界面不仅权限条目不全，而且解释比较粗略。这让很多普通用户摸不着头脑，而基于安装使用应用程序的目的下，用户通常会授予该应用权限，这也成了恶意软件可以大肆传播的主要原因。市场上的一些主流手机管家软件，如金山手机卫士、腾讯安全管家等，尽管提供了应用程序权限信息的查询功能，但还是比较粗糙且不一定准确。另外如果应用程序联网升级，则新版本可能会在更新中申请到新的权限，所以帮助一般用户了解各个应用所具有的权限信息、帮助他们进行判断选择，时时监控权限使用与升级情况变得十分重要。

图2 模拟器Android2.3.3中应用安装权限提示界面

3 权限检测系统架构

基于上述分析，本文提出了一种结合电脑端和手机端，能够对未安装的 APK 文件和已安装的 APP 应用程序进行深入权限检测的监测系统。该系统架构如图 3 所示分为两大模块：基于反编译的权限提取模块和基于 PackageManager 的权限提取模块。其中后一个模块按功能细化为两部分：按应用程序列表提取每个应用对应的权限，并给出权限列表；按一些敏感权限列表监测出所有使用该权限的应用程序，并给出程序列表。结合上述两个监测功能，能够帮助用户更深入更全面的了解和实时监控应用程序的权限信息，提高用户的安全意识，有利于保障系统的安全，保护用户隐私信息安全。

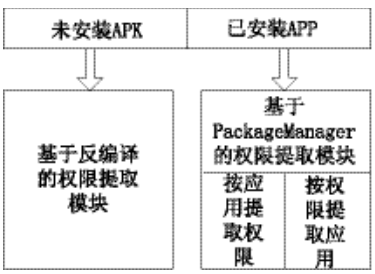

图3 检测系统架构

4 检测系统模块实现及测试

4.1 基于反编译的权限提取模块

每个 Android 应用程序的 APK 文件中都包含有一个 AndroidManifest.xml 文件，它记录着软件的一些基本信息，包括软件的包名、运行的系统版本、用到的组件、应用程序所申请的权限信息等。该文件以加密形式存储于 APK 文

件中，因此增加了监测提取权限信息的难度。

如果想通过获得 AndroidManifest.xml 文件来提取程序权限信息，则必须先要对 APK 文件进行反编译处理。这里采用了 Apktool 反编译工具。

命令语句为：apktool d<file.apk><dir>.

其中 <file.apk> 代表了要反编译的 APK 文件的路径，<dir> 代表了反编译后的文件的存储位置。

反编译成功后，会生成一系列目录与文件：其中 smali 目录下存放了程序所有的反汇编代码；res 目录则是程序中所有的资源文件；还有本模块所需要的声明权限信息的 AndroidManifest.xml 文件。图 4 中列出对图 2 中提示界面安装“单机版传奇”的 APK 文件进行反编译得到的 AndroidManifest.xml 文件中涉及权限声明部分。

图4 AndroidManifest.xml权限声明部分

从图 4 中可以看出，应有 11 条 uses-permission 标签，即程序发布者本意申请 11 条权限。但图中红框标出的字段明显出现了拼写错误，系统将只能识别出 10 条权限，这在下面的模块将给出对比结果。造成这样结果是由于系统无法识别此条权限申请，即使用户同意安装授权，应用程序也将不能拥有此条安装快捷方式的权限。也由此可见，在 Android 平台上发布应用程序能力是参差不齐，因此系统发布管理机制亟待改善。

4.2 基于PackageManager的权限提取模块

Android 系统为应用管理功能提供了大量的 API。根据功能的不同，这些 API 分为两大类：ActivitManager 相关和 PackageManager 相关 [8]。ActivityManager 相关类 API 是对运行时管理功能和运行时数据结构的封装，包括以下功能：激活 / 去激活 activity，注册 / 取消注册动态接受 intent，activity 生命周期管理等。PackageManager 相关类 API 是对所有基于加载信息的数据结构的封装，包括以下功能：安装、卸载应用，查询 permission 相关信息，查询 Application 相关信息 (application，activity，receiver，service，provider 及相应属性等)。

本监测系统中基于 PackageManager 的权限提取模块就是利用对 PackageManager 类的 API 的调用来实现权限获取。通过对上下文 context.getPackageManager（）来获取包管理服务；getInstalledPackages（）来获取所有应用包结构的集合；通过遍历集合中的每一个应用，调用 packageinfo.requestedPermissions 来获取到每个应用的权限信息；调用 packageinfo.packageName 来获取包名。再通过 packageinfo 来获得 applicationInfo 节点，调用其下的 appinfo.loadIcon 来获取应用图标，调用 appinfo.loadLabel 来获取程序应用名。在具体的系统呈现方面，本模块又可以划分为两个子模块：按应用提取权限模块和按权限提取应用模块。

按应用提取权限模块：本模块包括两个 activity，分别用来显示所有安装的应用程序列表和每一个应用程序详细的权限信息。第一个 activity 中，重写类 AppInfoProvider 中的 getAllApps（）函数方法，其中包括获取系统中所有应用程序包结构的集合，遍历系统所有应用程序的包信息，并通过 FLAG_SYSTEM 标签来识别是系统应用还是第三方应用，以此来决定是否予以列表显示。通过上文提到的 requestedPermissions 来获取权限信息，packageName 来获取包名，loadIcon 来获取应用图标信息，loadLabel 来获取应用程序名信息。呈现方面重写了数组适配器，定义了一个类 AppBrowerAdapter 继承 BaseAdapter 类，重写了 getView（）方法，负责把数据设置给 ListView 的每一个具体的小单元 app_item。item 包括应用程序图标、应用程序名、权限个数以及包名四个方面内容，以此列表方式显示出手机内所有的第三方应用程序。并在列表的每一个 item 上设置了监听器 setOnItemClickListener（），通过 getItemAtPosition（）来传递点击条目的数据，点击即启动第二个 activity，同样以 ListView 方式显示此应用的详细权限列表。此处为了方便普通用户查阅，采用了 Android 轻量级 SQLite 数据库，以 Android Developers permission 文档为基础，对其简要解释进行扩充和中文翻译，建立 database.db3 数据库文件。通过权限名查找到数据库中的权限条目，读取相应内容的方式给予显示每条权限详细、通俗易懂的解释，即使普通用户也很容易读懂每条权限的作用及可能造成的危害。

按权限提取应用模块：本模块同样包含两个 activity，

分别用来显示敏感权限列表和权限所属信息及包含本权限的所有应用。第一个 activity 中，使用 SimpleCursorAdapter 类配合 ListView 来显示，查询数据库文件辅以简单解释，以此来列出 16 个最易受恶意软件利用的敏感权限列表，包括拨打电话、发送短信、获取地理位置等保护等级为危险的权限。同样在每个 Item 上设置了监听器并重写 onItemClick（）方法以实现用户点击条目，启动第二个 activity 浏览具体的某一条权限的详细信息。调用 SQLite 数据库文件来给出这条敏感权限所属的权限组、保护等级和详细的描述。获取 PackageManager 中的程序包集合，根据某一条选定的敏感权限，遍历所有第三方应用程序包内的权限信息，若有匹配，则添加此程序包对应的应用，采用 ListView 给出列表，方便用户在觉察到手机异常时排查出可能的恶意软件。

4.3 测试与分析

实验测试环境：安卓模拟器 Android 2.3.3 上对开发的检测系统进行测试。

按应用提取权限模块：性能测试结果如图 5 所示。可以看到“单机版传奇”这个应用检测到了 10 条权限，与上文 APK 文件的检测结果相吻合。

图5 测试结果1

按权限提取应用模块：性能测试结果如图 6 所示。

图6 测试结果2

通过两大模块的测试结果可以看出，此系统可以准确给出所有第三方应用程序所拥有的权限列表，并在手机端设计了用户友好的界面，系统的可行性得到了证明。针对实例“单机版传奇”这个应用，通过反编译权限提取模块可以看到权限申请居然出现了低级的拼写错误，即使不是恶意软件，软件本身也会因为缺少这条权限而出现问题；再通过基于 PackageManager 权限提取模块，可以看到此应用名为单机版，但实际申请了包括访问网络、读取系统日志、获取地理位置等完全不应有的敏感权限。相信通过此监测系统提供的安全信息，普通用户也可以很容的判定此应用为恶意的广告或吸费软件而予以删除。

5 结束语

本文在分析了 Android 的安全机制的基础上，研究了系统粗糙简陋的权限提示界面及 Android 平台上发布应用程序能力是参差不齐等固有缺陷会带给用户使用的安全问题，提出了一种结合电脑端和手机端，对未安装的 APK 文件和已安装的 APP 应用程序进行深入权限检测的系统。该系统不仅能对权限做出详细、易懂的解释，并能提供用户排查基于某些敏感权限的应用程序，提供用户再警示、再判断的机会，提高使用者安全意识，保护使用者隐私及利益不受不明侵害的目的。下一步工作将考虑从 Linux 系统内核的改写上对应用程序所申请的某些敏感权限作出限制，如阻止某些应用连接互联网，发送短信等权限，真正断绝吸费软件恶意扣费，保障用户的财产和隐私不受侵害。（责编 吴晶）

参考文献

[1]Android Developers.http://developer.android.corn/guide/basics/what-is-android.html[EB/OL],2011-03-15.
[2] 新浪科技 . http://tech.sina.com.cn/it/2013-05-10/07478325514.shtml [EB/OL], 2013-05-10.
[3]DoNews. http://www.donews.com/net/201305/1495781.shtm [EB/OL], 2013-05-13.
[4] 靳岩，姚尚朗 .Google Android 开发入门与实战 [M]. 北京：人民邮电出版社 ,2009.
[5]Google.Android Reference:Manifest File-Permissions [EB/OL]. http://developer.android.com/guide/topics/manifest/manifest-intro.html
[6]Google. Android Reference: Security and Permissions [EB/OL]. http://developer.android.com/guide/topics/security/security.html
[7] 廖明华，郑力明 .Android 安全机制分析与解决方案初探 [J]. 科学技术与工程，2011，(09)：6350-6355.
[8] 红黑联盟 .Android 的 pachageManager 和 ActivityManager 的功能简介 M http://www.2cto.com/kf/201108/100493.html[EB/OL],2011-08-17.