

一种基于意图的安卓应用安全检测方法研究

刘 玮

(重庆师范大学 涉外商贸学院 数学与计算机系, 重庆 401520)

摘 要:针对目前 Android 移动应用市场受感染应用逐渐增多,应用软件安全性普遍较低,恶意软件检测难等问题,首先对 Android 应用的(显式和隐式)意图进行研究,接着对收集众多应用软件的权限和意图进行分析。通过对比 Android 应用权限发现,Android 应用意图信息是识别 Android 受感染应用程序的一个更有效的特性。然后研究了 AndroDialysis 框架利用意图检测恶意应用的方法和原理,最后通过对收集的 Android 应用意图信息进行实验,验证其有效性,实验达到了理想效果。如果将 Android 意图和安卓权限结合起来进行应用安全分析验证,则有可能进一步提高安卓应用的检测率。该研究能够为移动安全检测提供参考。

关键词:安卓应用;安全检测;意图;权限;AndroDialysis

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2019)05-0102-05

doi:10.3969/j.issn.1673-629X.2019.05.022

Research on a Method of Security Detection for Android Based on Intent

LIU Wei

(Department of Mathematics and Computer, School of Foreign Trade and Business,
Chongqing Normal University, Chongqing 401520, China)

Abstract:For the problem of increasing number of malware in current Android application market, the generally low security of application software and the difficulty in detecting malicious software, we first study the (explicit and implicit) intent of Android application, and then analyze the permission and intent information from many Android applications we collect. By comparison, we discover that intent is an effective feature to identify the Android infected application. Then we study the methods and principles of the AndroDialysis framework which is used to malware detection by analysis of the Android intent. Finally we collect the Android application intent information and do experiments to verify its effectiveness, and the experiment achieves ideal results. If Android intent and Android permission are combined to security analysis and validation, it would further improve the detection rate of Android applications. This study can provide reference for Android malware application detection in future.

Key words:Android application; security detection; intent; permission; AndroDialysis

0 引言

随着互联网技术和移动应用技术的发展,智能手机得到了快速普及。目前智能手机以其日益强大的计算能力、网络传输能力、功能繁多的应用能力极大地改变着人们的生活。Android 系统由于其开放、自由的特性成为恶意软件攻击的重要目标。目前,恶意软件正迅速渗透到主流的 Android 应用程序市场,引发了用户的广泛担忧^[1]。很多学者基于 Android 设备中的恶意软件检测进行了研究,主要有静态分析、动态分析以

及基于机器学习的混合检测三种方式^[2]。

进程通信是 Android 框架最显著的特征之一,该机制用于控制组件访问不同的敏感服务。在 Android 平台上,进程通信通常在后台由绑定消息的意图来驱动执行,意图提供了应用程序执行操作的抽象定义。文中对 Android 意图(显式和隐式)进行研究,通过研究发现意图是识别恶意应用程序的重要特性。相对于 Android 的其他特性(如权限^[3]),意图具有更丰富的语义特征,能够通过分析语义特征找出恶意软件^[4]。

收稿日期:2018-04-11

修回日期:2018-08-15

网络出版时间:2018-12-21

基金项目:重庆市教育科学技术研究项目(KJ1501702)

作者简介:刘 玮(1984-),男,讲师,硕士,研究方向为嵌入式软件开发。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20181221.1445.010.html>

1 Android 意图探索

意图是 Android 平台上一个复杂的消息传递机制,用于控制应用程序内部或者应用程序之间的安全访问。应用程序必须在 AndroidManifest. xml 里定义意图过滤器才具有使用特定意图的权限。

在 Android 应用程序中,意图可分为应用程序内 (intra-application) 通信和应用程序间 (inter-application) 通信。应用程序内通信是指一个 Android 应用的不同活动之间的通信。由于 Android 应用程序包含许多活动,应用程序进行交互时,用户可以从一个活动 (Activity) 跳转到另一个活动。Android 意图可以辅助完成活动执行并与用户交互,完成活动之间的数据推送,携带数据到指定活动。应用程序间通信实现了不同应用程序之间的通信,意图用于完成给其他应用程序发送信息或数据,并且接收从其他程序获取的数据或信息。为了控制接收指定意图,应用程序在 AndroidManifest. xml 须定义意图过滤器 (intent-filter)。应用程序之间的实际通信是通过绑定器 (Binder) 实现的。Binder 提供了一个执行环境和另一个环境之间的数据或功能绑定的功能。每一个 Android 应用运行在自己的 Dalvik 环境,绑定器为通信服务搭建桥梁^[5]。图 1 展示了应用程序间通信的架构。

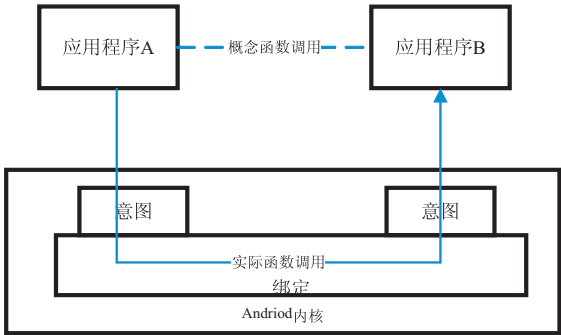


图 1 Android 应用程序通信架构

应用程序的部分读写区域由绑定器管理。当应用程序 A 向应用程序 B 发送消息时,绑定程序先在应用程序 B 的存储空间开辟接收缓冲区,然后从应用程序 A 的发送缓冲区复制消息,将消息发送到应用程序 B 的接收缓冲区,并告诉其消息接收的位置。最后接收方直接从该位置读取该消息。当应用程序 B 完成消息处理后,通知绑定器将该存储空间释放。

意图有两种类型:显式 (explicit) 和隐式 (implicit)。当设计人员确定执行指定操作的组件时,使用显式意图。此组件可以是任意的活动、服务或广播接收器。显式意图可以完成指定目标的程序内通信或者应用程序间信息交换。隐式意图不能指定确切的目标应用程序。为了解决这个问题,Android 提供了链接用户所有应用程序的列表。这个列表在 AndroidManifest.

xml 文件用意图过滤器过滤。意图有三个组成部分—动作 action、类别 category 和数据 data。动作用于描述意图要执行的行为,例如 MAIN (主程序入口)、CALL (打电话)、BATTERY LOW (电量低)、SCREEN ON (抓屏)以及 EDIT (编辑)等。category 是意图指定的类别,如发射器 LAUNCHER、浏览 BROWSABLE 和小工具 GADGET。数据为动作执行提供必要的的数据。例如,打电话需要电话号码,编辑需要指定文档或网址。表 1 展示了显式或者隐式意图的示例代码。隐式意图仅声明用 intent.action_view 来打开 URL,显式意图可明确指定使用 com.android.chrome 来打开。

表 1 Android 显意图和隐式意图

显式意图	隐式意图
<pre>String url = www.baidu.com; Intent explicit = new Intent (Intent.ACTION.VIEW); explicit.setData (Uri.parse (url)); explicit.setPackage (" com.android.chrome"); startActivity (explicit)</pre>	<pre>String url = www.baidu.com; Intent implicit = new Intent (Intent.ACTION.VIEW); implicit.setData (Uri.parse (url)); startActivity (implicit)</pre>

2 数据收集和分析

实验从 Android 应用市场收集很多主流的应用程序,这些应用软件既包括干净的应用程序,也包括受感染的应用程序。文中使用 Python 代码应用程序集中提取权限和意图。表 2 列出了干净和受感染应用的 10 个最常用权限。

在表 2 中,有 5 种权限是干净和受感染应用都经常使用的:INTERNET, WRITE_EXTERNAL_STORAGE, WAKE_LOCK, ACCESS_COARSE_LOCATION 和 READ_PHONE_STATE。在受感染的应用中,请求 SEND_SMS, RECEIVE_SMS 和 READ_SMS 权限是危险的权限,WRITE_SMS 也是属于使用频繁的危险权限,该权限排名第 11,经常被 22% 的受感染应用程序请求调用。因此推断访问 SMS 相关权限功能是应用受感染的关键证据。30% 的应用程序请求 ACCESS_FINE_LOCATION 权限获得精确位置,32% 的应用程序请求 ACCESS_COARSE_LOCATION 权限访问最接近的位置,这两个权限的目标类似。因此在一般情况下,应用软件是否受感染可以通过权限来度量。文中也提取了应用程序的意图,如表 3 所示。该表显示了干净和受感染应用的 10 种最常用意图的使用频率 (由于所有的安卓应用都要应用 VIEW 意图,所以该意图忽略)。

表 2 干净和受感染应用的 10 个最常用权限

安全应用		危险应用	
权限	频率/%	权限	频率/%
INTERNET	98	INTERNET	98
ACCESS_NETWORK_STATE	89	READ_PHONE_STATE	89
WRITE_EXTERNAL_STORAGE	83	WRITE_EXTERNAL_STORAGE	67
WAKE_LOCK	53	SEND_SMS	54
READ_PHONE_STATE	53	RECEIVE_SMS	38
ACCESS_WIFI_STATE	48	WAKE_LOCK	38
GET_ACCOUNT	42	READ_SMS	37
VIBRATE	41	ACCESS_COARSE_LOCATION	32
BILLING	39	ACCESS_FINE_LOCATION	30
ACCESS_COARSE_LOCATION	24	READ_CONTACTS	23

表 3 干净和受感染应用的 10 个最常用意图

干净的应用		受感染的应用	
意图	频率/%	意图	频率/%
SEND_MULTIPLE	45	BOOT_COMPLETED	56
SCREEN_OFF	23	SENDTO	45
USER_PRESENT	18	DIAL	42
SEARCH	17	SCREENOFF	37
PICK	10	TEXT	28
DIAL	9.5	SEND	27
GET_CONTENT	9	USER_PRESENT	22
EDIT	8.7	PACKAGE_ADDED	21
MIDEA_MOUNTED	8	SCREEN_ON	18
BATTERY_CHANGE	7	CALL	10

恶意应用通过调用意图 BOOT_COMPLETED 来启动恶意活动;意图可使用 CALL 或 DIAL 拨打电话。在安卓应用中 CALL 需要拨打电话的权限,而 DIAL 不需要此权限。因此在恶意应用中调用意图 DIAL 比调用 CALL 多很多,也意味着恶意应用程序在没有用

户知情的情况下可以越权拨打电话。

3 移动恶意软件检测系统研究

AndroDialysis 是一种检测 Android 应用的框架,该系统通过分析隐式和显式意图检测 Android 应用是否安全^[6]。图 2 是 AndroDialysis 系统的体系结构。该检测系统分为四个子模块:反编译、数据提取、智能学习和决策。检测系统通过图形接口将结果反馈给用户。

3.1 反编译

反编译用于将 APK 文件解码。其中,Android Manifest.xml 是一个加密文件,需要解码使其可读。而 dex 文件是由 java 源文件编译的,可在 Dalvik 虚拟机上执行字节码文件^[7]。可以使用 apktool 反编译获取。反编译模块用于生成可读的 AndroidManifest.xml 和 Smali 版本的 java 代码^[8]。

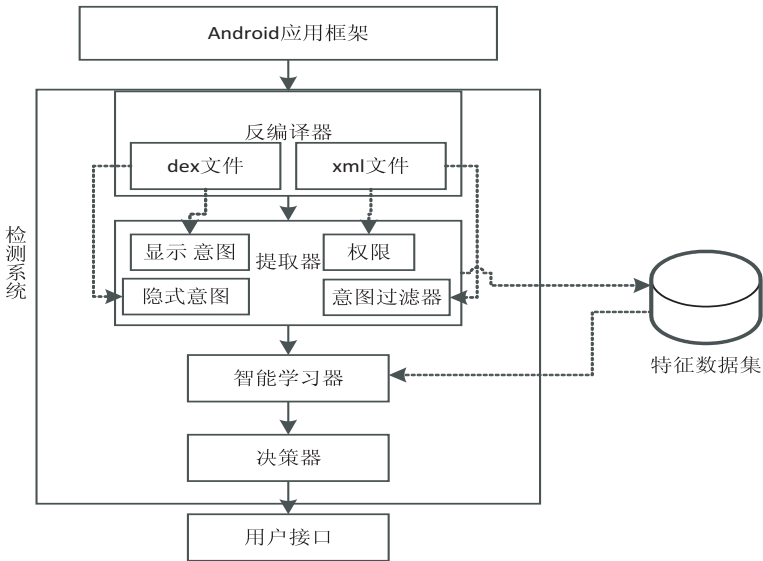


图 2 AndroDialysis 系统的体系结构

3.2 提取器

提取器从 java 代码和 AndroidManifest. xml 中提取显式意图,隐式意图,意图过滤器以及权限。Python 语言的 beautifulsoup 包能够从 AndroidManifest. xml 文件中获得意图过滤器和权限。使用 Androguard 来翻译 dex 文件,可获取(隐式或显式)意图^[8]。将提取后的数据存储在特征数据库中用于后续操作。

3.3 智能学习

智能学习模块读取特征数据库中的内容,利用贝叶斯网络算法通过学习数据进行行为分析,然后向决策子模块输出模型^[9]。贝叶斯网络算法已经能够用于解决现实世界中的人脸识别问题。该算法是一种对偶过程算法,首先学习网络结构,再学习概率表。贝叶斯网络算法根据局部评分指标来学习网络结构的数据^[10]。为了计算局部得分,贝叶斯网络采用搜索算法。网络数据学习完成后,贝叶斯网络利用估计器学习概率表。上述两个步骤定义如下:

假设 $V = \{X_1, X_2, \dots, X_k\}$ 是一个变量集合,且 $k \geq 1$ 。贝叶斯网络 Bs 是一个由条件概率 V 构成的有向无环图(DAG)。存在一个概率表 $BP = \{P(\nu | P\alpha(\nu)), \nu \in V\}$,其中 $P\alpha(\nu)$ 是 BS 中 V 的父集。贝叶斯网络可以将概率分布化简为:

$$P(\nu) = \prod_{\nu \in V} P(\nu | P\alpha(\nu))$$

相比其他算法,贝叶斯网络具有运算速度快、算法开销低,易于建立专家和学习系统并集成到系统中,用法较为成熟,易于优化等特点。

3.4 决策

决策模块负责确定数据干净或者危险。该模块分别接收提取器和智能学习获得的两组数据集。其中一组数据集通过智能学习建立模型,然后利用该模型检测从提取器接收到的数据;而另一组数据来源于提取器,该数据全部从 Android 应用中提取。决策模块利用智能学习建立的模型来检测应用程序的危险程度,

并将最终结果传递给用户模块。

4 实验结果分析

4.1 根据意图分析攻击

从安全的角度,分析意图数据集并评估意图的状态和特征信息来获得攻击是可行的。隐式意图由于不能指定目的组件,通常提供给接收某类意图的实体。所以发送隐式意图时,不能确保该意图被目的接收组件接收。恶意应用程序可通过在 AndroidManifest. xml 文件中声明一个意图过滤器(intent-filter)来拦截该意图及其所带的动作、数据及分类信息(actions, data, categories)。这种未经授权的意图接收会使恶意应用程序获得对任何匹配的意图数据访问,使活动劫持。

在已获得的数据中,受感染 Android 应用声明意图过滤器比干净的应用多 7.5 倍。平均每个干净的应用声明 1.2 个意图过滤器,而每个受感染的应用程序则声明 1.7 个意图过滤器。很明显受感染的应用更倾向于使用意图过滤器(intent-filter)来拦截意图,从而拦截这些活动。因此开发人员应尽量使用显式意图指定接收目标,这样可以有效避免恶意应用程序劫持活动。

4.2 实验结果

通过在手机上运行实验,检测该方法的有效性和效率。系统使用 6.0.1 棉花糖版本 Android 操作系统,设备的 RAM 4 GB 的存储空间 32 GB。

贝叶斯网络是一种计算局部得分指标的搜索算法和概率表学习相结合的估算方法。简单估算结果分别使用了 K2^[11], Geneticsearch^[12], HillClimber^[13], LAG-DHillClimber^[14] 四种搜索算法。为了获得最佳结果,实验采用了不同配置。实验结果由正确率(或称检测率 TPR)和错误率(FPR)组成。实验通过多次迭代获取,并且随着迭代次数的增加,系统分析数据更精确,结果更理想。图 3 显示了每次迭代后实验的正确率和

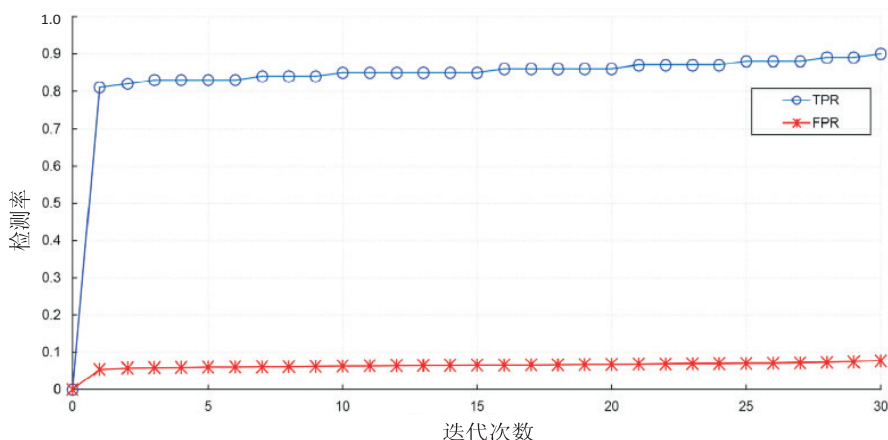


图3 迭代次数与检测率的关系

错误率。

当迭代次数增加时,正确率由 80% 增加到 90%,错误率却增长缓慢,仅从 6% 增加到 9%,所以迭代次数越多,结果越准确。根据前面分析可知,Android 意图是安卓恶意应用检测的一个有效的属性。

尽管意图的特性并不是最终的解决方案,但是如果将意图和其他已知特性如权限结合使用,就能准确对 Android 恶意应用进行评估。文中通过分析目前主流安卓应用程序,发现利用 Android 意图对恶意软件检测率为 91%,大于使用安卓权限检测的 83%。若结合安卓意图和权限安卓恶意检测率会更高^[15]。

5 结束语

作为战略性新兴产业的代表,移动互联网的出现促进了社会信息化水平的进一步提高,移动应用的进步也使得人们的生活更加便利。解决安卓应用安全性问题对于保证国家安全、个人隐私等具有至关重要的作用。文中主要针对 Android 应用软件的安全问题进行分析,探索了意图在 Android 应用检测中的重要特性,研究了 Android 应用安全检测的框架 - Andro Dialysis,分析并验证意图是受感染应用检测的一个有效特性。通过结合安卓意图和安卓权限可以加强 Android 应用安全检测的正确率。总体而言,目前安卓市场上应用安全普遍存在,软件安全防范技术和体系处于早期建设阶段,依然存在很多问题,需要投入更多精力促进其进一步发展。

参考文献:

[1] 卿斯汉. Android 安全研究进展[J]. 软件学报,2016,27(1):45-71.

[2] TONG Fei, YAN Zheng. A hybrid approach of mobile malware detection in Android[J]. Journal of Parallel and Distributed Computing,2017,103:22-31.

[3] WANG Wei, WANG Xing, FENG Dawei, et al. Exploring permission includes risk in Android application for malicious application detection[J]. IEEE Transactions on Information Forensics and Security,2014,9(11):1869-1882.

[4] 杨天长,崔浩亮,牛少彰,等. Android 应用 Intent 通信风险分析及检测[J]. 北京理工大学学报,2017,37(6):625-630.

[5] 唐俊杰. 面向 Android 系统中 Intent 通信机制的漏洞分析框架及其应用[D]. 济南:山东大学,2017.

[6] FEIZOLLAH A, ANNAL N B, SALLEH R, et al. Andro-Dialysis: analysis of Android intent effectiveness in malware detection[J]. Computers & Security,2017,65:121-134.

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

[8] 刘方圆,孟宪佳,汤战勇,等. 基于 smali 代码混淆的 Android 应用保护方法[J]. 山东大学学报:理学版,2017,52(3):44-50.

[9] REHMAN Z U, KHAN S N, MUHAMMAD K, et al. Machine learning-assisted signature and heuristic-based detection of malwares in Android devices[J]. Computers and Electrical Engineering,2018,69:828-841.

[10] 曹杰. 贝叶斯网络结构学习与应用研究[M]. 合肥:中国科技大学,2017.

[11] CHEN Xuewen, ANANTHA G, LIN Xiaotong. Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm[J]. IEEE Transactions on Knowledge & Data Engineering,2008,20(5):628-640.

[12] YAN L J, CERCONE N. Bayesian network modeling for evolutionary genetic structures[J]. Computers & Mathematics with Applications,2010,59(8):2541-2551.

[13] CHICKERING D, GEIGER D, HECKERMAN D. Learning Bayesian networks: search methods and experimental results [C]//Proceedings of the fifth conference on artificial intelligence and statistics. Florida, USA: IEEE,1995:112-128.

[14] SALEHI E, GRAS R. An empirical comparison of the efficiency of several local search heuristics algorithms for Bayesian network structure learning[C]//Proceedings of the learning and intelligent optimization workshop. [s. l.]: [s. n.],2009:1-14.

[15] SOKOLOVA K, PEREZ C, LEMERCIER M. Android application classification and anomaly detection with graph-based permission patterns[J]. Decision Support Systems,2017,93:62-76.