

Android 应用模拟交互技术的研究

邹加磊,洪 亮,杨鸣坤

(西北工业大学 自动化学院 控制与网络研究所,陕西 西安 710072)

摘 要:针对于行为分析的 Android 恶意代码检测技术均需要收集大量的运行数据,文中提出了一种基于用户模拟交互技术的数据收集方式。使用计算机及若干个 Android 设备,自动化完成有效分析数据的收集,减少了人工参与的程度。通过结合使用制定的 Android 应用程序的运行策略,包括环境差异策略、时间差异策略、事件差异策略,以及测试工具等,完成应用程序在 Android 设备上的自动安装、交互运行、数据收集以及自动卸载。验证实验表明,该用户模拟交互技术能有效收集应用程序的运行数据,是一种可行的应用方案。

关键词:Android;恶意程序;模拟交互;自动化

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2014)11-0032-03

doi:10.3969/j.issn.1673-629X.2014.11.008

Research on Android Application Simulation Interactive Technology

ZOU Jia-lei, HONG Liang, YANG Ming-kun

(Control & Network Institute, School of Automation, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Large amount of data is needed by the Android malware detection technology based on behavior analysis. A way of data collection based on the technology of user simulation interaction is proposed. Using a computer server and several Android devices, can automatically collect useful data for analysis and reduce the work of human involvement. By combing the designed simulation strategy which includes environment, time, events difference and testing tools, the system can automatically accomplish application automatic installation, interaction running, data collection and automatic uninstallation. Experimental results show that the proposed solution architecture is feasible, which can effectively collect the data of program.

Key words: Android; malware; simulation interaction; automation

0 引 言

Android 是一种基于 Linux 的自由开放源码的操作系统,主要应用于便携设备,如智能手机和平板电脑等。截止 2013 年第二季度,IDC 发布的最新报告显示:Android 在全球智能手机操作系统的占有率已经高达 79.3%,远超其他智能系统。

然而伴随着 Android 占有率的不断提高,Android 平台上的安全问题也越来越严重。网秦公司《2012 安全报告》指出:2012 年,有超过 3 200 万台 Android 设备被感染,被感染设备数量仍在不断上升中,恶意软件的横行给用户带来极大的损失。近年来,越来越多的机构和个人参与到 Android 恶意软件的研究中来,很多采用的是基于行为的动态分析,故需要收集大量的必要数据,如用户使用过程中程序调用函数的情况等。

文中提出了一种 Android 应用程序的模拟交互技

术的分析方法,通过制定一定的策略,运用自动化工具,使用计算机完成模拟用户使用 Android 设备过程中的数据收集,从而为进一步的行为分析做准备。

1 相关研究

目前,Android 恶意软件已经成为了信息安全的一个研究重点,研究的主要方法可以分为静态研究和基于行为分析的研究两种。基于行为的分析方法一般要求收集必要的应用程序信息,如应用程序所申请权限情况、使用网络情况以及运行过程中的函数调用、资源使用情况等,国内外有越来越多的机构和个人参与到这些研究中来。

国防科学技术大学的方志鹤对恶意代码进行了分类研究^[1]。北卡罗莱纳州立大学的 Yajin Zhou 等人通

过 Android Malware Genome Project^[2]进行了两年多的恶意代码收集,并对收集到的恶意代码进行了归类和特征分析^[3-4]。Wook Shin 等人研究了 Permission 的授权管理控制模型^[5]。W. Enck 等人提出了一个轻量级的不安全应用程序识别方法,其基于应用程序的 Permission^[6]。L. Davi 分析了权限扩展模型。David Barrera 等人则采用统计学的方法分析了应用程序中 Permission 的使用情况^[7]。张中文等人探讨了 Permission 机制与 Android 的安全^[8],丁丽萍则从系统角度进行了分析^[9],Shabtai 及 Iker B 等人提出了基于行为监测恶意软件的模型和方法^[10-11]。

在自动化测试工具方面,北京邮电大学的马红素等人,实现了一个简单的 Android 应用的自动安装、启动和运行系统^[12];南开大学的王伟,设计并实现了自动分析 Android 病毒的系统^[13]。

2 模拟交互的设计

模拟交互所需实现的就是为了让计算机能够模拟出用户使用 Android 设备的情况,并在运行过程中通过有效的监控,收集所需的数据。例如较多应用中所需要收集的底层函数调用信息,其评价目标包括自动化程度、交互的智能程度等。

2.1 设计目标

模拟交互实现的目标如图 1 所示。

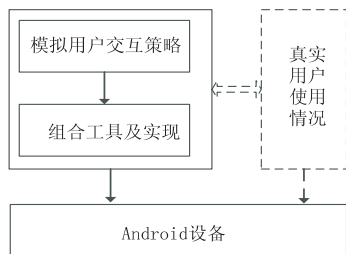


图 1 模拟交互效果图

图中,左侧框图为用户模拟的交互部分,在 Android 设备上,左侧模拟交互的实现完全依赖计算机的自动运行,实现接近真实用户使用 Android 设备的情况。相关运行数据的获取可以在此过程中截获,以便下一步的分析。其中,交互部分通过制定一定的用户交互策略,并结合使用一定的自动化测试工具等,模拟完成用户触摸屏幕、滑动 Trackball、按键及不同的方式组合等操作,触发应用程序的相关事项,从而达到模拟真实用户使用情况的效果。

2.2 实现策略

考虑到现实生活中,真实用户在使用手机时候的不同情况,诸如用户在不同的网络环境下使用 Android 设备应用:Wifi 环境、3G/2G 网络环境、无网络连接环境;用户在不同的时间内使用应用的情况;如某个应用

使用 1 分钟、10 分钟等不同时间,以及诸如微信等应用,随机后台长时间运行;用户使用应用程序特定的部分功能;还有单个应用前后台运行,不同应用间的交叉使用等情况,该交互设计共制定了三种差异策略。

环境差异策略:这里所指的环境,是 Android 设备的使用环境,真实用户在使用 Android 设备时,会有不同的网络环境、不同的设备使用环境(如电池使用情况)、不同的设备接入环境(如是否连接电脑,有无蓝牙连接等)。理想的用户模拟交互行为应该能模拟出以上场景,为实现自动化,以上环境可以使用脚本程序,调用 Android 设备提供的接口函数来实现。

时间差异策略:针对单个或者多个应用程序,用户在使用的时候,会有不同的时间使用情况,应用程序可以短时间或长时间使用,也可以间隔使用,可以后台长期运行,也可以和其他的应用程序交叉使用。为此,计算机在进行用户模拟交互时,应按照有差异的时间策略来运行应用程序,如特定时间触发特定程序、使得应用程序后台运行,几个应用程序按时间交替使用等。

事件差异策略:真实用户在使用 Android 设备时,会有侧重地使用其中的某些功能,为此,计算机在进行用户模拟交互时,可以触发不同的事件流,如利用 Monkey 工具发送事件流时,可以指定不同的事件类别、不同的包等。

如图 2 所示,某个待测试应用有可能应用其中的单个策略,也有可能应用不同的组合策略,并在相应的策略下,提取相应的调用信息,从而完成模拟用户的交互运行信息收集。当然模拟用户的程序使用行为是一个复杂的过程,需要后期通过实验进行不断的反馈,并进行相应的调整,从而更好地提高模拟用户行为的真实性,保证提取信息的有效性。

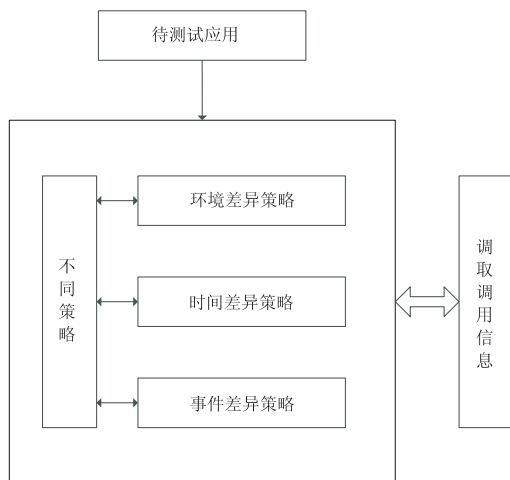


图 2 模拟交互策略图

2.3 关键技术

用户模拟交互技术的有效性依赖于模拟策略的合理性。部分恶意软件仅在特定条件下,才会展示出其

恶意行为,这就需要模拟交互能够提供合适的环境,较多的场景和较长时间的测试,会增大其恶意行为暴露的可能性,但会降低有效数据的比例,这就需要通过后期的不断反馈及进一步的研究来校正交互的策略模型。另外,测试方法也需要不断地进行拓展,所以下一步需要做的工作还有很多。

3 原型的实现与验证

3.1 原型实现

根据上述模拟交互的设计原理,文中实现了一个简单的自动化收集数据的软件,其使用 Google 提供的 Android 虚拟机模拟 Android 设备,使用联想台式机 (Intel Pentium 3 GHz,5 GB,500 GB) 充当服务器,使用 Google 提供的 Monkey 工具来发送随机事件数据流,运用 adb 实现数据的通信,同时,使用 strace 工具完成底层 Linux 调用序列的抓取。

简化的模拟交互实现图如图 3 所示。

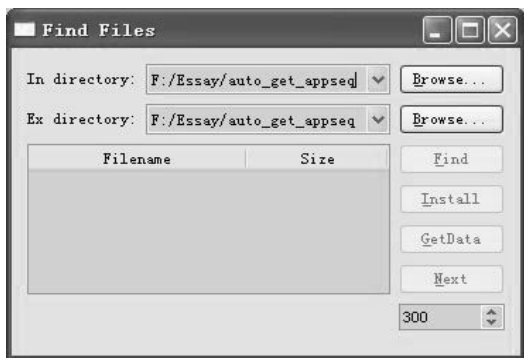


图 3 简化的模拟交互实现图

由图 3 可见,该交互设计的简单实例可以实现 Android 应用的自动安装、自动启动、自动模拟交互和自动运行并收集相应的数据,最终将数据导入到指定位置,其具体流程如图 4 所示。

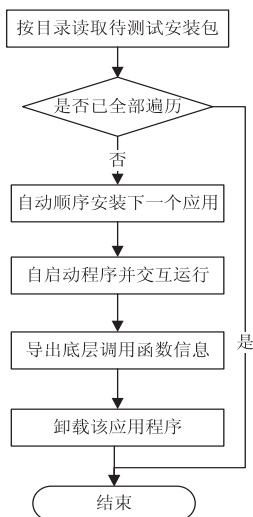


图 4 交互运行收集数据总体流程图

首先,将要测试的应用程序放到某个目录下,之

后,使用该软件读取该目录,目录读取完后,将自动进行第一个应用程序的安装,安装完毕后,由 Monkey 发送一个单一事件,即应用启动事件。程序启动后,开始运行此应用程序,同时监控该进程的底层 Linux 调用信息。需要指出的是,文中仅实现了按照一定时间间隔发送数据流,之后仍有很多差异化运行实现的工作要做。待应用程序运行一定时间后,发送进程终止信号,停止交互运行,之后将底层函数调用信息导出到指定目录,最终卸载该程序,从而完成了一个简单的交互模拟运行,并收集到了相关的数据。

3.2 原型验证

此次验证的应用程序为“Android Malware Genome Project”中选取的 AnserverBot 家族中的 10 个应用程序,并对每个应用程序其进行了 5 分钟运行,对底层 Linux 调用函数序列收集。

如图 5 所示,通过对数据进行分析,发现其在 Linux 调用函数的种类及调用频率较高的函数具有一定的共性。

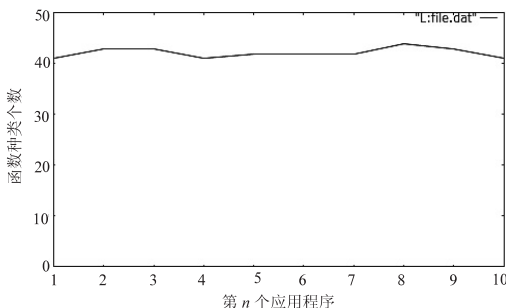


图 5 AnserverBot 调用函数种类总数图

由图 5 可见,10 个应用程序调用到的 Linux 函数种类介于 41 ~ 44 个之间。同时,调用频率较高的几个函数均为 recvfrom、clock_gettime、ioctl、sigprocmask、futext、mprotect 等,分析其原因,因为其均具有类似的功能。当然,判断应用程序的恶意与否需要收集更多的信息,并对数据进行更深层次的分析,如函数调用的序列之间的关系等,这也是分析工作所需要做的。

4 结束语

伴随着 Android 设备越来越普及,恶意软件的蔓延也给用户带来了极大的损失。基于行为的分析方法可以更好、更及时地检测出恶意行为,但需要大量的有用信息支撑。而大量数据的处理,需要结合使用自动化的运行工具。文中提出的用户模拟交互行为的设计,是一种有效的数据收集方式,通过简单实例的验证,说明了该交互行为的实用性。

交互模拟行为的差异策略,需要考虑较多的情况,并根据结果进行不断地反馈矫正,故下一步需要做的

3 结束语

文中系统在内部某中心试运行一个月以后,共接收了 3 125 次下载请求,这其中包含了 871 次重复下载,系统实际下载 2 254 篇论文。节省了约 27.87% 的下载量。另外系统在使用过程中积累的访问数据和下载数据后期可以用来做数据挖掘,用于分析机构内学术趋势和学术状况,这对于科研机构的长期发展也有一定的实际意义^[14-15]。

系统暂时只覆盖了部分学术资源库,后期需要增加学术资源的覆盖面。系统中保存了很多有价值的使用数据,如何有效地利用这些数据挖掘出有价值的信息将是这个系统在推广使用过程中下一步值得思考和解决的问题。

参考文献:

- [1] 李育娣. 文献检索中提高查全率与查准率的方法探讨[J]. 图书馆学研究,2002(11):92-93.
- [2] 李纪欣,王 康,周立发,等. Google Protobuf 在 Linux Socket 通讯中的应用[J]. 电脑开发与应用,2013,26(4):1-5.
- [3] 詹恒飞,杨岳湘,方 宏. Nutch 分布式网络爬虫研究与优化[J]. 计算机科学与探索,2011,5(1):68-74.
- [4] 尹 江,尹治本,黄 洪. 网络爬虫效率瓶颈的分析与解决方案[J]. 计算机应用,2008,28(5):1114-1116.
- [5] Stevanovic D, An Aijun, Vlajic N. Feature evaluation for Web

(上接第 34 页)

工作包括继续提高差异策略的精细化,模拟交互效果的真实性,数据收集的有效性等。

参考文献:

- [1] 方志鹤. 恶意代码分类的研究与实现[D]. 长沙:国防科学技术大学,2011.
- [2] Android malware genome project[EB/OL]. 2010. <http://www.malgenomeproject.org/>.
- [3] Zhou Yajin, Jiang Xuxian. Dissecting Android malware: characterization and evolution[C]//Proc of 2012 IEEE symposium on security and privacy. San Francisco: IEEE Computer Society, 2012: 95-109.
- [4] 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 19th annual network and distributed system security symposium. [s. l.]: [s. n.], 2012.
- [5] Shin W, Kiyomoto S, Fukushima K, et al. A formal model to analyze the permission authorization and enforcement in the Android framework[C]//Proceedings of the 2010 IEEE second international conference on social computing. Minneapolis: IEEE, 2010: 944-951.
- [6] Enck W, Ongtang M, McDaniel P D. On lightweight mobile phone application certification[C]//Proceedings of the ACM

crawler detection with data mining techniques[J]. Expert Systems with Applications, 2012, 39(10): 8707-8717.

- [6] 朱庆生, 邹景华. 基于本体论的论文检索[J]. 计算机科学, 2005, 32(5): 172-173.
- [7] 邓志鸿, 唐世渭, 张 铭, 等. Ontology 研究综述[J]. 北京大学学报(自然科学版), 2002, 38(5): 730-738.
- [8] Abruscia V M, Fouqueré C, Romanoa M. Formal ontologies and coherent spaces[J]. Logic Categories Semantics, 2014(1): 67-74.
- [9] Yan Wei, Zanni-Merkb C, Cavalluccia D, et al. An ontology-based approach for inventive problem solving[J]. Engineering Applications of Artificial Intelligence, 2014, 27: 175-190.
- [10] van Ruijven L C. Ontology for systems engineering[J]. Procedia Computer Science, 2013, 16: 383-392.
- [11] Liu Chilun. Cloud service access control system based on ontologies[J]. Advances in Engineering Software, 2014, 69: 26-36.
- [12] Simonm. HtmlAgilityPack's documentation[EB/OL]. 2006. <http://htmlagilitypack.codeplex.com/>.
- [13] 万维网联盟(W3C). XPath 语法介绍[EB/OL]. 2010. <http://www.w3school.com.cn/xpath/>.
- [14] 李 洁, 丁 颖. 语义网、语义网格和语义网络[J]. 计算机与现代化, 2007(7): 38-41.
- [15] 李 蕾, 郭祥昊. 基于语义网络的概念检索研究与实现[J]. 情报学报, 2000, 19(5): 525-531.
- conference on computer and communications security. Chicago: ACM, 2009.
- [7] Barrera D, Kayaclik H G, van Oorschot P C, et al. A methodology for empirical analysis of permission-based security models and its application to Android[C]//Proceedings of the 17th ACM conference on computer and communications security. Chicago: ACM, 2010.
- [8] 张中文, 雷灵光, 王跃武. Android Permission 机制的实现与安全分析[C]//第 27 次全国计算机安全学术交流会论文集. 出版地不详; 出版者不详, 2012.
- [9] 丁丽萍. Android 操作系统的安全性分析[J]. 信息安全, 2012(3): 28-31.
- [10] 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.
- [11] Burguera I, Zurutuza U, Nadjm-Tehrani S. Crowdroid: behavior-based malware detection system for Android[C]//Proc of 1st ACM workshop on security and privacy in smartphones and mobile devices. Chicago: ACM, 2011: 15-26.
- [12] 马红素, 郭燕慧. Android 应用自动化动态测试工具的研究及实现[D/OL]. 2012. <http://www.paper.edu.cn>.
- [13] 王 伟. Android 病毒行为自动分析工具的设计与实现[D]. 天津: 南开大学, 2012.

Android应用模拟交互技术的研究

作者: [邹加磊](#), [洪亮](#), [杨鸣坤](#), [ZOU Jia-lei](#), [HONG Liang](#), [YANG Ming-kun](#)
作者单位: [西北工业大学 自动化学院 控制与网络研究所, 陕西 西安, 710072](#)
刊名: [计算机技术与发展](#) 
英文刊名: [Computer Technology and Development](#)
年, 卷(期): 2014(11)

本文链接: http://d.wanfangdata.com.cn/Periodical_wjfz201411008.aspx