Android 恶意软件检测方法研究

冯 博,戴 航,慕德俊
(西北工业大学 自动化学院,陕西 西安 710072)

摘 要:针对 Android 恶意软件泛滥的局面,提出了一种基于行为的恶意软件动态检测的方法。首先,综合收集软件运行时的动态信息,包括软件运行时系统的信息和软件的内核调用信息,并将内核调用序列截断成定长短序列的形式。其次,将各方面信息统一为属性、属性值的形式。以信息增益作为指标,选用 C4.5 算法筛选出信息增益高、作用不重叠的属性,并依据信息增益的大小为各属性正比分配权重因子。最后,用 K 最近邻算法完成机器学习,识别出与样本类似的恶意软件,并将未知类型的软件标记为疑似恶意。实验结果表明,该方法识别率高、误报率低。通过增大学习样本库,识别的效果可以进一步提高。

关键词:Android 安全;恶意软件;动态检测;机器学习
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2014)02-0149-04
doi:10.3969/j.issn.1673-629X.2014.02.036

Research of Malware Detection Approach for Android

FENG Bo,DAI Hang,MU De-jun
(School of Automation,Northwestern Polytechnical University,Xi'an 710072,China)

Abstract:In view of the flood situation for Android malware,propose a method of behavior-based dynamic malware detection. First,get a comprehensive collection of software run-time information,including system information and kernel calls. The kernel call sequences are truncated to fixed length. Second,form all the information as property and values. Taking information gain as an indicator,select properties that have high information gain and different impact by applying the C4.5 algorithm,and proportionally assign weighting factor to properties based on the size of the information gain. Finally,apply K-Nearest Neighbor algorithm to complete the process of machine learning,making the system identify malicious software that similar to the sample,and regard unknown types of software as suspected malware. The result of experiment shows that the method has a high true positive rate and low false positive rate. Moreover,the result can be further improved with the increase of the learning sample library.

Key words:Android security;malware;dynamic detection;machine learning

0 引 言

Android 系统以其开源、免费的特性,使它在当今智能手机市场上的占有率已高达 75%。Android 应用开发门槛低,使用 Android 手机的人数多,为它开发的应用软件种类繁多、数量巨大。由于 Android 软件市场对 Android 应用软件安全性没有进行有效的审查,针对 Android 平台的恶意软件十分泛滥[1-2]。因此,及时且有效地检测出恶意软件,让用户安全地使用 Android 手机十分必要。当前,有关 Android 恶意软件检测的研究处于探索阶段,没有成熟、完善、实用的技术[3-4]。该研究领域发展的一个显著特点是研究者将 PC 机上恶意代码的研究方法和理论移植到 Android 平台上[5]。难点在于要适应 Android 平台硬件资源相对匮乏的情况。在众多检测方法中,基于行为的检测方法是研究的热点。然而,现有的一些检测方法设计得比较简单,有的只进行了特征行为的匹配[6],有的只统计内核调用的频次[7]。而且这些方法都没有充分考虑到在实际应用中需要克服恶意软件升级、变种的问题。在测试时,通常只是应用了少量的样本来检验所用的检测方法是否能够识别恶意软件,然后就得出了相关结论。文中提出了一种综合分析软件运行时的各方面信息,并采用 K 最近邻算法进行分类

收稿日期:2013-03-28 **修回日期**:2013-06-30 **网络出版时间**:2013-11-29
基金项目:2012 教育部博士点基金(20126102110036)
作者简介:冯 博(1989-),男,湖南人,硕士研究生,研究方向为网络与信息安全;戴 航,硕士生导师,研究方向为网络与信息安全;慕德俊,博士生导师,研究方向为控制理论、网络信息安全。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20131129.1020.053.html

识别的 Android 恶意软件检测方法,可以有效地克服上述问题。

1 技术背景

研究者通过调研大量恶意软件的样本,总结恶意软件的行为特点,给出了很多建设性的结论[8]。他们将恶意行为分为如下 4 类:

1)窃取用户的信息,例如个人资料、金融凭据、位置信息等;

2)收发增值服务类的消息;

3)推送广告,方式多种多样,例如发送广告消息,通过搜索服务推送广告;

4)恶作剧,编写者出于娱乐心态制作的软件。

恶意软件申请的权限与良性程序有着很大的区别,例如:是否具备 SMS 的使用权限可以作为一个重要判别标准,11 个恶意软件中有 8 个(73%)申请了发送 SMS 消息的权限,而只有 4% 的良性软件申请了该权限[9]。

通过解析 Android 软件转码静态地获取软件行为,会耗费大量的运算资源,并且不能真实地反映软件行为。研究者借鉴 PC 机上检测恶意代码研究的成功经验,采用的基于行为的动态检测方法已经取得了初步的成果。研究者通常选择 Linux 内核调用作为软件行为的表征。分析内核调用序列的方法主要有三种:

1)直接将内核调用与实际操作对应起来;

2)将内核调用序列看成一个向量进行分析;

3)分析内核调用的频率。

Android 的 Linux 内核有超过 300 个内核函数调用,有研究者对哪些内核调用更能反映软件行为特征做了相关的分析工作[6]。应用该成果去除冗余的内核调用,将大大减少分析数据的计算量。

2 检测方法的设计与实现

动态检测系统由三个部分构成:动态信息采集模块、数据预处理模块、数据分析模块。动态信息采集的目标是获取到的信息能够全面地反映软件运行的情况[10]。动态信息分为两个方面:一是软件本身的行为信息,二是软件运行时的系统环境信息。采集到的信息繁杂,形式多样,内核调用序列的数据量巨大。数据预处理具体是指,将系统状态信息以及软件的特征行为形式化、量化处理,内核调用表中包含了很多无关的内核函数调用,将其去除,再将序列截断并统计各定长短序列的频次。经过数据预处理后,数据在形式上已经表现为可分析的二维表的形式。数据分析模块的工作分三个方面:

1)精简二维表的表项,去除作用小、冗余的属性;

2)计算属性的权值;

3)应用机器学习算法学习分类规则,学习完成后识别未知程序。

图 1 为该检测系统的处理流程图。在实际检测中存在偏离良性,又不能认定为恶意的软件,将其交由专业技术人员对其开展进一步的分析,然后再反馈给系统,提高系统应对恶意软件的能力。

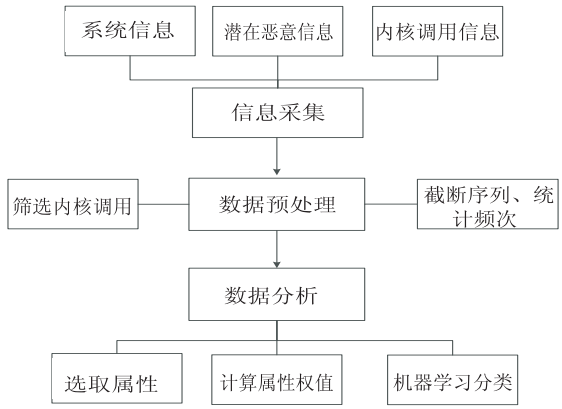

图 1 处理流程

2.1 数据采集

将软件运行时的情况完整地描述出来,是检测分析工作的基础。数据采集模块拥有三个监控器,分别采集三方面的信息。

1)系统信息监控器。系统级的信息可以直接读取获得,具体收集的内容有:系统的运行状态(活跃、休眠),数据网络状态,通话状态,Wi-Fi 状态,电池使用状态,软件对 CPU 的占用率。

2)潜在恶意行为监控器。由研究数据可知[9],恶意软件存在一些常见的行为,这些行为可以认为是潜在恶意的行为。应用程序接口层为软件提供打包好的服务,监控器设置在应用程序接口层上,能直观地监测软件的行为。监测的内容包括:使用网络的情况,SMS 消息使用情况,访问用户个人信息。

3)内核调用监控器。软件所有的行为最终都是通过调用内核函数来实现的,它能够反映软件行为的特点。它对检测恶意软件的意义已经在 PC 机上得到了证明。在 Android 平台上获取软件的内核调用序列十分方便,通过内核提供的调试工具包 ADB(Android 调试桥)可以直接截获进程内核调用序列。

2.2 数据预处理

数据采集模块得到的信息不能直接应用于算法分析,此节将介绍数据预处理的过程。三个监控器中的内核调用监控器采集到的数据量很大,以运行五分钟为例,软件的内核调用数量从几百到上万条不等,图 2 为音乐播放器产生的内核调用的一部分。首先,经过研究分析,诸如进程调度、内存管理、信号管理、消息管理等类型的内核调用对于检测软件是否安全的意义不

大,筛除它们可以减少分析计算的工作量,也让数据更具有针对性。表 1 是保留下的内核调用的列表,总共选取了 27 个关键的内核调用,它们能够反映软件行为的基本特征。其次,从图 2 中可以看到,内核函数执行时的输入参数和返回的结果意义不明确,可以进行精简,保留内核调用名称即可。

```
msgget(0x1, 0xbebef900, 0, 0xa812124c)      = 0
ioctl(20, 0xc0186201, 0xbebef750)           = 0
msgget(0x1, 0xbebef900, 0, 0xa812124c)      = 0
recv(1099333636, 0x1, 2147483647, 0)        = 0
msgget(0x3, 0xbebef850, 0, 0xa812124c)      = 0
ioctl(20, 0xc0186201, 0xbebef6d8)           = 0
msgget(0x1, 0xbebef900, 0, 0xa812124c)      = 0
```

图 2　内核调用数据片段

表 1　保留的内核调用

类别	内核调用名称
进程管理	clone,execve,fork,vfork,capget,capset,getuid,getuid32,geteuid,geteuid32
文件 I/O	accept,fcntl,bind,connect,mkdir,open,read,recv,rename,rmdir,send,stat,unlink,write
其他	sysinfo,access,uname

内核调用序列去除冗余后,依旧是一条长度很大的序列;需要对其进行截断才能进一步分析处理。在实验中,选取定长序列的长度为 5,截断的方法不是简单的分割,而是每切割一次,只向前进一位。例如,长度为 100 的长序列,就截断成了 96 个定长为 5 的短序列,然后统计定长序列的种类以及各定长序列的频次[11]。系统信息监控器和软件行为监控器采集到的信息多为描述性、离散的信息,将它们变成属性、属性值的形式就可以将它们和内核调用整合到一起分析处理。属性值将根据各自的取值范围与取值方法确定,具体方法在下一小节会给予说明。表 2 为三类信息统一形式的方法。

表 2　二维表的形式

属性名	属性值
系统信息	状态值
潜在恶意行为名	有/无
定长序列名	频次

2.3　数据分析

在上一步的处理中,可以察觉到几个问题。第一,选取了 27 个内核调用,由这 27 个排列组合而成的定长短序列的种类数量高达 14 348 907 个,中间有很多无意义的、重复的属性。第二,不同的属性识别软件类型的能力不一样,各属性在计算距离时要增加权重因子。属性对软件类型的区分力的强弱体现在信息增益上,可以用如下的方法来实现。使用 C4.5 决策树算法[12],能够处理连续数值型属性的分裂问题,因此定长序列的频次就不需要特殊处理了。通过计算可以挑选出信息增益高的属性,剪除掉那些信息增益少、作用重复的属性。属性确定后,根据各个属性的信息增益值,按照同等比率给予权重。

信息增益的计算方法:$\mathrm{Info}_A(D)$ 代表经过 A 这样的一次划分,要达到完整信息 $\mathrm{Info}(D)$ 还需要多少信息,$\mathrm{Gain}(A)$ 是经过 A 方式划分所得到的信息,即信息增益。

$$\mathrm{Info}(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$\mathrm{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \mathrm{Info}(D_j)$$

$$\mathrm{Gain}(A) = \mathrm{Info}(D) - \mathrm{Info}_A(D)$$

经过上述的各个步骤,软件的行为特征已经完全概括成二维表(属性、属性值)的形式,一个软件对应一张二维表,用 K 最近邻算法对各二维表进行学习计算。K 最近邻算法将相似度转换为距离,行为相似的软件,二者距离较近,会聚成一类。在系统学习样本的阶段,分类结果出来后,可以对照样本实际分类的情况,调整属性的数量,以及权重分配的方法。完成学习后,识别出既偏离良性,又偏离恶意的软件,可以提交给专业技术人员人工分析,再将结果反馈给系统,建立新的类别。

3　测试分析

Yajin Zhou 等收集了 1 260 个样本[1],从他们收集的样本中取出 65 个作为恶意软件。在 Android 市场上下载 50 个良性的软件,将这 115 个软件作为实验的样本,将各软件进行分类整理。其中挑选 50 个恶意软件、35 个良性软件作为系统学习的样本,其余 30 个作为测试样本使用。实验的硬件平台选用的手机型号为 Motorola me 525,手机上搭载 Android 2.3.5 系统。为确保运行环境完全真实,在手机上录入虚假的个人信息,并配备一张可正常使用的 SIM 卡。

经过实验检测:15 个良性软件,其中有 13 个判断正确,2 个作为疑似恶意软件提交;15 个恶意软件,13 个判断正确,1 个判断错误,1 个作为疑似恶意软件提交。将疑似恶意软件的类型反馈给系统后,系统即拥有了对它们的识别能力。

从结果来看,该检测方法已经表现了良好的性能,由于学习的样本数量有限,判别为疑似恶意软件的较多,这样的情况应该可以通过大量样本的学习来改善。

4　结束语

通过全面地收集软件运行时的动态信息,将各类

信息整合成二维表(属性、属性值)的形式,以信息增益作为挑选属性和分配权重因子的标准,应用机器学习的算法让系统进行学习,使系统对恶意软件形成了良好的识别能力。需要进一步探究的是:

1)内核调用的选取方法,在该方法中,是通过分析恶意行为在底层内核调用的实现方法,确定保留哪些内核调用;其科学性有待严谨的实验证明;

2)Android 平台恶意软件的动态检测系统的实现要充分考虑到它的特殊性,既要保证能在本地执行检测,又要尽量少地占用硬件资源。

参考文献:

[1] Zhou Yajin, Jiang Xuxian. Dissecting Android malware: Characterization and evolution[C]//Proc of IEEE symposium on security and privacy. [s. l.]: [s. n.], 2012.

[2] Schmidt A D, Schmidt H G, Batyuk L, et al. Smartphone malware evolution revisited: Android next target[C]//Proc of 4th IEEE international conference on malicious and unwanted software. [s. l.]: [s. n.], 2009.

[3] 符易阳,周丹平. Android 安全机制分析 [J]. 信息网络安全,2011(9):23-25.

[4] 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,2011,11(26):6350-6355.

[5] Shabtai S, Kanonov U, Elovici Y. "Andromaly": A behavioral malware detection framework for Android devices[J]. Journal of intelligent information systems, 2012, 38: 161-190.

[6] Isohara T, Takemori K, Kubota A. Kernel-based behavior analysis for Android malware detection[C]//Proc of seventh international conference on computational intelligence and security. [s. l.]: [s. n.], 2011.

[7] Burguera I, Zurutuza U, Nadjm-Tehrani S. Crowdroid: Behavior-based malware detection system for Android[C]//Proceedings of the 1st ACM workshop on security and privacy in smartphones and mobile devices. New York: [s. n.], 2011.

[8] Felt A P, Finifter M, Chin E, et al. A survey of mobile malware in the wild[C]//Proceedings of the 1st ACM workshop on security and privacy in smartphones and mobile devices. New York: [s. n.], 2011.

[9] Felt A P, Greenwood K, Wagner D. The effectiveness of application permissions[C]//Proc of USENIX WebApps. [s. l.]: [s. n.], 2011.

[10] 路 程. Android 平台恶意软件检测系统的设计与实现[D]. 北京:北京邮电大学,2012.

[11] Han K S, Kang B. Malware classification using instruction frequencies[C]//Proc of ACM symposium on research in applied computation. [s. l.]: [s. n.], 2011.

[12] Firdausi I. Analysis of machine learning techniques used in behavior-based malware detection[C]//Proc of advances in computing, control and telecommunication technologies. [s. l.]: [s. n.], 2010.

(上接第 148 页)

论还存在着不足之处,这将作为下一步工作的重点。并针对大规模分布式应用的特殊需求,进一步优化、扩展该方案,使得该方案能够在大规模分布式应用中在安全性与效率上有更好的表现。

参考文献:

[1] 宁 葵. 访问控制安全技术及应用[M]. 北京:电子工业出版社,2005.

[2] 王连强,张 剑,吕述望,等. 一种基于密码的层次访问控制方案及其分析[J]. 计算机工程与应用,2005,41(33):7-10.

[3] 陈 原,王育民,肖国镇. 公钥密码体制与选择密文安全性[J]. 西安电子科技大学学报(自然科学版),2004,31(1):135-139.

[4] 王圣宝,曹珍富,董晓蕾. 标准模型下可证安全的身份基认证密钥协商协议[J]. 计算机学报,2007,30(10):1842-1852.

[5] 曾梦岐,卿 昱,谭平璋,等. 基于身份的加密体制研究综述[J]. 计算机应用研究,2010,27(1):27-31.

[6] 苏金树,曹 丹,王小峰,等. 属性基加密机制[J]. 软件学报,2011,22(6):1299-1315.

[7] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//Proceedings of the 13th ACM conference on computer and communications security. Alexandria, VA, USA: [s. n.], 2006: 89-98.

[8] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]//Proc of IEEE symposium on security and privacy. Oakland, California, USA: [s. n.], 2007: 321-334.

[9] ITU-T Rec. X509(2000) | ISO/IEC 9594-8:2000, The Directory: Public-key and attribute certificate framework[S/OL]. 2000. http://www.iso.org/iso/store.htm.

[10] ITU-T Rec. X509(2005) | ISO/IEC 9594-8:2005, The Directory: Public-key and attribute certificate framework[S/OL]. 2005. http://www.iso.org/iso/iso_catalogue/-catalogue_tc/catalogue_detail.htm?csnumber=43793.

[11] 中华人民共和国信息产业部. GB/T 16264.8-2005,信息技术开放系统互连目录第 8 部分:公钥和属性证书框架[S]. 北京:中国标准出版社,2005.

[12] OASIS Standard, eXtensible Access Control Markup Language (XACML) Version2.0[S/OL]. 2005. http://www.oasis-open.org/committees/xacml.

Android恶意软件检测方法研究

作者： 冯博， 戴航， 慕德俊， FENG Bo， DAI Hang， MU De-jun
作者单位： 西北工业大学 自动化学院,陕西 西安,710072

刊名： 计算机技术与发展

ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2014(2)