



专题:基于 Android 系统的终端安全

基于组合式算法的 Android 恶意软件检测方法

陈昊¹, 卿斯汉^{1,2,3}

(1. 北京大学软件与微电子学院, 北京 102600;

2. 中国科学院软件研究所, 北京 100190;

3. 中国科学院信息工程研究所信息安全国家重点实验室, 北京 100093)

摘要:为解决当前恶意软件静态检测方法中适用面较窄、实用性较低的问题,通过组合式算法筛选出最优分类器,并以此为基础实现了一个检测系统。首先使用逆向工程技术提取软件的特征库,并通过多段筛选得到分类器的初步结果。提出了一种基于最小风险贝叶斯的分类器评价标准,并以此为核心,通过对初步结果赋权值的方式得到最优分类器结果。最后以最优结果为核心实现了一个 Android 恶意软件检测系统原型。实验结果表明,该检测系统的分析精度为 86.4%,并且不依赖于恶意代码的特征。

关键词:恶意软件检测;特征选择;组合式算法;最小风险贝叶斯评价;危险权限组合

中图分类号:TP301

文献标识码:A

doi: 10.11959/j.issn.1000-0801.2016253

Android malware detection method based on combined algorithm

CHEN Hao¹, QING Sihan^{1,2,3}

1. School of Software and Microelectronics, Peking University, Beijing 102600, China

2. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

3. State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing 100093, China

Abstract: In order to solve the problems in applicability and usability of today's static malware detection method, a detection system was implemented by using the optimal classifier selected by a combined algorithm as the core. Firstly, the reverse engineering was used to extract the software feature, then the preliminary results of the classifier was got by multi-stage screening. A classifier evaluation was presented based on minimum risk Bayes. Using the new one as the core, the optimal classifier results was got by assignment. Finally, an Android malware detection system prototype was realized using the optimal results as the core. Experimental results show that the analysis accuracy of the proposed detection system was 86.4%, and does not depend on characteristics of the malicious code.

Key words: malware detection, feature selection, combined algorithm, minimum risk Bayes evaluation, dangerous permission combination

收稿日期:2016-07-30;修回日期:2016-09-28

基金项目:国家自然科学基金资助项目(No.61170282)

Foundation Item: The National Natural Science Foundation of China(No.61170282)



1 引言

随着移动网络的快速发展,Android 平台以开放、便利、免费等特点逐渐走进了人们的日常生活。但与此同时,其蕴含的巨大经济利益也吸引了众多非法牟利者,恶意软件种类和数量也与日俱增^[1],用户的隐私和财产安全受到了严重威胁。

当前针对 Android 平台上的恶意软件检测方法主要分为两类:基于动态行为和基于静态分析的检测。前者将程序置于可完全监控的环境下安装运行,通过分析程序运行中的行为与系统参数对程序进行判断,包括但不限于移动设备事件^[2]、访问的关键系统应用^[3]、敏感数据的流向^[4]、系统的资源消耗^[5]等内容。这类方法分析结果较为直观,但有时容易出现遗漏。后者则不安装运行程序,通过逆向工程技术提取软件的特征,使用机器学习方法判断软件是否具有恶意行为。此类方法分析结果对已知恶意特征代码有很高的检测率,但对于加密、变形的恶意代码效果不佳。

静态分析检测也可以分为两大类:一类不使用任何随机性算法,如静态数据流分析^[6];另一类则倾向于特征提取与特征选择。特征选择可以看作一个寻找最优的问题。对于大小为 n 的特征集合,搜索空间由 2^n-1 种可能的状态构成。已经证明,搜索最优特征子集是一个 NP 难问题^[7]。考虑到可行性问题,在实际工作中通常找到次优解即停止。该方向研究中常见的算法有信息熵方法^[8]、最近邻算法^[9,10]、朴素贝叶斯^[10]、支持向量机^[11]等。根据与学习算法的组合方式,这些算法一般可分为 Filter 和 Wrapper 两类模型算法^[12]。但鉴于两类模型算法的互补性,许多研究者提出将两类算法结合使用以获得更优解。如 Huang^[13]提出的两段组合式特征选择算法,Galar 等人^[14]提出的针对不平衡分类的混合算法等。这些方法都能在结果上较传统算法取得一定进步,但目前这方面工作普遍存在一些难以解决的问题,如特征子集规模划定、针对不同样本库的通用性等。

借鉴上述相关的研究成果,在此基础上进行了一定的扩展。首先,在特征选择过程中不局限于某一种特定的算法,而是通过多种算法共同寻找最优分类器。其次,所实现的组合式算法并非是在一个算法的结果上继续删减,而是以该结果为准,试图通过赋值对该结果进行优化;最后,结合传统分类器评价标准与自行给出的新标准,以综合评价的分类器结果为核心实现了一个检测系统原型。

2 恶意软件特征概述

根据中国互联网协会公布的官方定义,恶意软件具有以下特点:强制安装、难以卸载、浏览器劫持、广告弹出、恶意收集用户信息、恶意卸载和恶意捆绑^[15]。恶意软件通常包含两种能力:一是攻击与破坏的能力;二是生存和自我保护能力。

恶意软件的攻击有比较明确的目标,根据攻击目标的不同,实际的软件特征也有差异。尽管存在一些“多功能”的恶意软件,如 Android 木马 Geinimi 同时实现了远程控制、信息窃取、短信拦截、静默拨号、静默安装等多种攻击方式^[16],但在特定攻击方式上可被检测系统针对性地进行检测。恶意软件的生存能力则是指实现以下一种或多种的功能,包括干扰基于特征的检测、干扰对代码的分析、干扰对恶意程序的检测与干扰安全软件正常工作等。为达到以上目的,恶意软件通常会采取多种技术进行自我保护,如多态、混淆和加密。如在 2013 年出现的 Android OBAD 木马,其本身 dex 文件中的所有字符串都被加密,并且代码被混淆,使得常规的检测方式很难生效^[17]。

但无论恶意软件自身采用何种手段,有几个性质是普遍存在的:恶意软件通常会要求正常软件所需数倍的权限,有些可能直接造成用户损失的权限组合也在被申请之列;恶意软件的关键特征与正常软件差距较大,如在代码中隐藏恶意网址、SP(服务提供商)号码、大量进行反射调用等;恶意软件使用加密、加壳等保护手段的概率比正常软件大很多。尽管部分开发者宣称是为了保护自身关键技术而采用的防护措施,但该方式并不利于软件为普通用户所接受。

综上,可以考虑以下内容作为恶意软件的判定依据:软件是否存在危险权限组合;软件是否存在可疑字符串,如 SP 号码或未知 URL 等;软件是否存在敏感函数调用或大量异于正常软件的函数调用。在此基础上,选择提取待测软件的静态声明权限、常量与系统 API 调用作为该软件的初始特征集。

3 检测系统框架设计

为保证最后得到的检测系统具有良好的可维护性,将检测系统框架分为如下两个部分:第一部分为预处理,主要工作包括样本收集、特征提取与选择、最优解寻找等模块,最终得到一个分类器结果;第二部分则是面向用户的

交互,以第一部分最终获得的分类器结果为评价核心,对用户输入的待测样本进行评价,最终给出评估报告。当系统进行更新时,只需更新两部分共用的分类器结果即可。基本流程框架如图 1 所示。

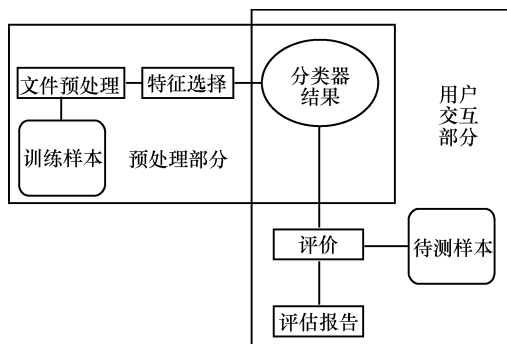


图 1 检测系统框架结构

整个系统分为预处理和用户交互两大部分。在预处理部分,首先将训练样本逆向成为 smali 代码,然后对所有的 smali 文件进行代码分析与特征提取,得到样本的初始特征集。在特征选择步骤中,对全部样本的特征进行汇总,然后通过不同的特征选择算法得到该算法情况下的分类器。最后通过 Wrapper 类算法对分类器进行权重赋值,结合分配器评价标准找到最优的分类器结果。预处理部分对用户是不可见的,并且对外输出仅有分类结果,因此易于维护和结果更新。

在用户交互部分,检测系统需要实现与用户的交互界面并接收待测样本。该部分采用与前一部分相同的方式对样本进行特征提取,并以上一步得到的分类器结果为评价核心对样本进行评价。模块化结构使得系统具有很高的可维护性。在以分类器结果评价的基础上,再加入如权限组合评测模块等内容,最终共同生成评估报告。对于用户而言,可见的仅有交互界面与最终的评估报告,可以保证良好的用户体验。

4 特征选择与结果分析

本文实验样本共使用从网络上搜集的 Android 软件安装包共计 1 313 个,其中正常样本 1 000 个,恶意样本 313 个。考虑到市场上正常软件与恶意软件所占比例不同的情况,实验中对于每一种算法将采用全部样本、1:1、5:1、10:1 共 4 种抽样比例进行算法学习,并采用多次重复后投票机制保留最终结果。每次投入实验的样本数量均为规定比例与现有样本总量基础上可能取得的样本数量最大值。

4.1 初步规则筛选

实验首先将所有样本进行逆向与特征提取。经统计,存在于全部正常样本当中的不同特征项共计 160 余万项,恶意样本中不同特征项共计 29 653 项。数据规模的庞大是由于在提取过程中,对所有的函数调用都当作特征进行处理造成的。因此首先需要对这庞大数量的特征进行初步筛选。

设定筛选规则如下:在正常样本中出现比例大于在恶意样本中出现比例的特征项排除。该类特征会对特征选择算法产生不利影响;排除在不同样本中出现频数低于样本总数 1% 的特征项,此规则可以有效排除某样本独有的函数;特征项名称内含有特殊字符的特征项排除(如“*”、“\$”等,下划线除外)。这一部分特征项是 smali 代码中相当于调用子类的函数,属于函数内部调用,因此不影响实验结果。所有特征经过初步规则筛选后,剩余有效特征项 3 377 项。

4.2 特征选择算法筛选

经过初步筛选后,判定为有效的特征仍然超出可接受的规模。为此采取两个步骤:首先选择较为适合大量数据筛选的特征选择算法将特征规模进一步缩小,然后再使用较为精确的算法对结果进行优化。为了确保最终结果较优,考虑采用多种特征选择算法进行上述工作。在各类特征选择算法中,Filter 模型算法比较适合对大量数据进行粗糙筛选^[18],因此第一步工作交由 Filter 模型中的信息熵相关算法和 K 近邻相关算法共同完成。最终实验选择的方法为信息增益、交叉熵、标准 Relief 以及参数分别为 5 和 10 的 ReliefF 算法。实验利用这 4 种算法与不同的样本抽样比例,总计得到 20 种初步分类器结果。

在第二步工作中,选择通过 Wrapper 模型中的遗传算法对分类器初步结果进行权重赋值,而非继续精简分类器规模。事实上,传统组合式算法得到的分类器结果中各项特征权重相同,即本方法筛选后所有权重均为 1 的特例;对于分类器结果的再度精简也只是本方法筛选后一部分权重设为 0 的特例。将所有特征项权重设为 1 的分类器加入算法的初代个体中,可以保证结果不劣于原始结果。

4.3 最优结果搜寻与结果分析

4.3.1 基于 ROC 曲线的评价标准

ROC 曲线(receiver operating characteristic curve,受试者工作特征曲线)是当前较为通用的一种评价标准^[19]。其评价方式可简单叙述如下:对于一个二分问题,首先给定



以下 4 个定义:真正类(true positive, TP)表示原本恶意的样本被划定为恶意的数目;假负类(false negative, FN)表示原本恶意的样本被划定为正常的数目;假正类(false positive, FP)表示原本正常的样本被划定为恶意的数目;真负类(true negative, TN)表示原本正常的样本被划定为正常的数目。进而引出以下两个定义:

- 真正类率(true positive rate, TPR): $TPR = TP / (TP + FN)$
- 负正类率(false positive rate, FPR): $FPR = FP / (FP + TN)$

以 FPR 为横坐标, TPR 为纵坐标, 根据阈值的变化可以得到一条曲线。在理想情况下, 所有样本分类正确, 此时 $FN=FP=0$, $FPR=0$, $TPR=1$, 在图中即为点(0,1)。因此曲线越接近上方, 可以视为曲线的判定准确率越高, 通常用曲线下方面积作为定量标注, 该面积即为 AUC (area under curve, 曲线下面积)值。采用 ROC 曲线的 AUC 值作为分类器评价标准, 实验首先对上一步得到的 20 种不同分类器进行评价, 即默认分类器内部的特征项同等重要时, 得到分析结果见表 1。

表 1 无权重情况下的分类器 AUC 值

算法/样本	全部样本	样本 1:1	样本 5:1	样本 10:1
IG	0.637 7	0.650 2	0.667 4	0.617 5
CE	0.514 3	0.630 9	0.466 2	0.535 5
Relief	0.590 6	0.529 9	0.608 8	0.550 1
ReliefF(5)	0.561 8	0.557 0	0.583 4	0.564 6
ReliefF(10)	0.585 3	0.530 8	0.598 0	0.584 1

为了能够得到更优的结果, 实验采用遗传算法对 20 个不同分类器分别进行赋值计算。设定参数如下: 最大进化代数 $T=50$, 随机生成个体数 $M=100$, 个体评价标准采用 ROC 曲线的 AUC 值, 下一代的生成规则为: 原有 100 个个体原样置入新个体集; 随机抽取原有 2 个个体进行交叉运算, 得到 400 个新个体置入新个体集; 随机抽取原有个体进行变异, 得到 250 个新个体置入新个体集; 随机生成 250 个新个体置入新个体集。然后从新个体集的 1 000 个个体当中应用评价标准, 选择出评价价值最高的 100 个个体进入下一代。最终得到结果见表 2。

表 2 有权重情况下的分类器 AUC 值

算法/样本	全部样本	样本 1:1	样本 5:1	样本 10:1
IG	0.689 1	0.701 5	0.707 8	0.654 0
CE	0.522 8	0.630 9	0.468 5	0.545 6
Relief	0.651 3	0.576 7	0.664 9	0.597 8
ReliefF(5)	0.610 8	0.594 9	0.638 2	0.611 8
ReliefF(10)	0.647 6	0.557 4	0.649 3	0.638 3

从表 2 可以看出, 在使用遗传算法进行分类器优化后, 信息增益算法依然具有更优的 AUC 值。

4.3.2 基于最小风险贝叶斯的评价标准

ROC 曲线与 AUC 值是近年来比较通用的一种用以度量分类模型好坏的标准, 但是这一标准也有其局限性。当 $AUC=0.5$ 时, 模型被认为等同于随机猜测, 没有预测价值, 这一信息默认的情况是正常样本与恶意样本数量相当。而在实际工作中, 正常样本数量会远大于恶意样本, 所造成的不平衡数据集在应用 ROC 曲线时会有一定的问题; 此外, 由于当前 Android 平台环境的特殊性, 模型中 FN 与 FP 所造成的损失并不相同, 因此使用统一的横纵坐标度量并非上策。

一个比较适合现状环境的模型是最小风险贝叶斯模型^[20]。在最小风险贝叶斯模型当中, FN 与 FP 所造成的损失是不同的。若将 FN 设定为标准损失 1, 那么 FP 则是一个大于 1 的数 λ 。对于一个确定的特征集与其对应的特征值构成的组合, 每一个样本在该组合下所对应的评价是确定的, 假设可以预先得知 λ 的值并根据 λ 的值来划定分类标准, 那么随着 λ 的变化, 使得总损失最小的划定标准也可能随之变化。简单地讲, 如果 $\lambda=1$, 此时判断正确与错误损失相当, 划定的标准就是使分类精度最高的那个标准; 如果 λ 趋近于无穷大, 那么任何一个 FP 类带来的损失都是不可估量的, 所以划定的标准就是使 FP 为零时 TP 能取得最大值的划分方式。即阈值为恶意样本中的最小权重。

考虑这样一条曲线: 横坐标 $x=\lambda/p$, 其中 p 为正常样本数; 纵坐标 $y=cost/\lambda$, 其中 $cost$ 为 λ 确定时总损失的最小值, 称这条曲线为定损曲线。当 $\lambda=1$ 时, $x=1$, $y=(1-\text{分类精度}) \times \text{样本总数}$; 当 λ 趋近于无穷大时, $FP=0$, $cost$ 为定值, 曲线为固定损失代表的一条水平线。一般的定损曲线如图 2 所示。

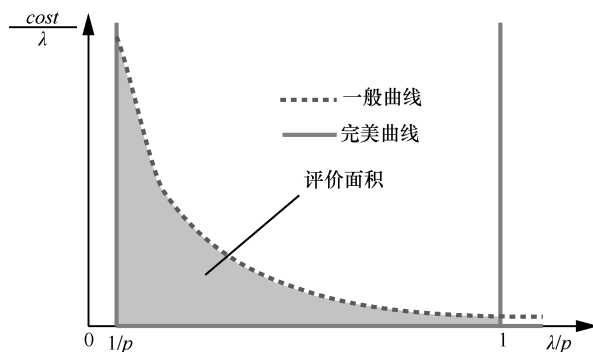


图 2 定损曲线示意

对于一个分类器而言,如果该分类器是完美的,那么定损曲线应与横坐标重合。一个分类器所对应的定损曲线越接近横坐标,说明该分类器对于不同 λ 的适用性越好。实验以横坐标区间 $[1/p, 1]$ 之间定损曲线与横轴围成的面积(图 2 中阴影部分)作为评价标准,对 20 个无权重分类器进行计算,并与 AUC 值结果进行对比,结果见表 3。

表 3 无权重情况下两种标准精度对比

样本比例	算法	AUC	SUF	精度
全部样本	IG	0.637 7	5.984 8	0.865 1
	CE	0.514 3	5.997 4	0.861 3
	Relief	0.590 6	6.104 9	0.815 6
	ReliefF(5)	0.561 8	6.038 1	0.844 6
	ReliefF(10)	0.585 3	5.944 3	0.846 9
样本 1:1	IG	0.650 2	1.753 2	0.722 0
	CE	0.630 9	1.683 7	0.814 6
	Relief	0.529 9	1.772 7	0.659 7
	ReliefF(5)	0.557 0	1.713 6	0.712 4
	ReliefF(10)	0.530 8	1.755 7	0.714 0
样本 5:1	IG	0.667 4	5.618 6	0.893 3
	CE	0.466 2	5.609 6	0.901 6
	Relief	0.608 8	5.659 5	0.860 0
	ReliefF(5)	0.583 4	5.614 3	0.867 5
	ReliefF(10)	0.598 0	5.625 2	0.877 5
样本 10:1	IG	0.617 5	4.979 2	0.940 0
	CE	0.535 5	4.983 2	0.941 8
	Relief	0.550 1	5.156 1	0.918 1
	ReliefF(5)	0.564 6	4.983 0	0.934 5
	ReliefF(10)	0.584 1	4.993 0	0.931 8

从表 3 可以看出,定损曲线面积与 AUC 值的增长趋势大致相反,与系统精度增长趋势相反,定损曲线面积越低的分类器评价效果越好。对这 20 个无赋值分类器继续使用与之前相同参数的遗传算法进行最优解寻找,得到结果见表 4。

从表 4 可以看出,在以定损曲线与横轴围成的面积作为分类器评价标准时,信息增益算法得到的分类器依然为

表 4 有权重情况下的分类器定损曲线面积值

算法/样本	全部样本	样本 1:1	样本 5:1	样本 10:1
IG	5.896 6	1.725 2	5.482 8	0.992 9
CE	5.935 4	1.683 7	5.595 9	4.876 0
Relief	6.008 9	1.757 3	5.480 1	5.037 3
ReliefF(5)	6.003 6	1.680 2	5.460 9	4.859 8
ReliefF(10)	5.825 0	1.753 2	5.464 5	4.867 1

最优。事实上,信息增益算法偏重于强调单特征对样本的影响,而 Relief 类算法偏重于多个特征组合对样本的影响。实验在判定恶意样本时,通过关键函数与权限特征进行判定的情况更多一些,因此信息增益算法得到的分类器结果较优。将上述结果以精度为标准进行比较,得到结果见表 5。

由表 5 可以看出,采用曲线面积作为指标的分类器最终结果在精度上优于传统 AUC 值指标的结果,因此实验选择以该类分类器结果作为检测系统的核心使用。

5 权限组合模块

考虑到在算法中并未对权限进行单独处理,检测系统还需额外添加危险权限组合检测模块。通过结合现有的权限功能说明与危险等级划分,本文将权限归于以下几个级别。

(1)破坏级

拥有该级别权限的应用可以通过调用特定的函数对系统直接进行危害。危害的定义不仅包括直接造成系统不能正常工作,也包括对用户能直接造成的其他损失。这一类权限包括:

- android.permission.BRICK
- android.permission.MASTER_CLEAR
- android.permission.MOUNT_FORMAT_FILESYSTEMS
- android.permission.REBOOT
- android.permission.SYSTEM_ALERT_WINDOW
- android.permission.REORDER_TASKS

表 5 有权重情况下的两种标准精度对比

样本	全部样本		样本 1:1		样本 5:1		样本 10:1	
	AUC 精度	SUF 精度	AUC 精度	SUF 精度	AUC 精度	SUF 精度	AUC 精度	SUF 精度
IG	0.864 4	0.864 4	0.722 7	0.706 0	0.844 6	0.849 2	0.790 5	0.859 8
CE	0.795 8	0.794 3	0.830 1	0.830 1	0.858 3	0.859 1	0.858 3	0.792 8
Relief	0.761 6	0.849 9	0.745 6	0.767 7	0.775 3	0.779 8	0.776 0	0.779 1
ReliefF(5)	0.785 9	0.787 5	0.788 2	0.853 0	0.782 9	0.848 4	0.788 2	0.789 0
ReliefF(10)	0.779 1	0.789 0	0.787 5	0.855 2	0.780 6	0.785 2	0.784 4	0.853 7



android.permission.INSTALL_PACKAGES
android.permission.CHANGE_COMPONENT_ENABLED_
STATE

android.permission.BROADCAST_STICKY
android.permission.CALL_PHONE
android.permission.CALL_PRIVILEGED
android.permission.DELETE_PACKAGES
android.permission.PROCESS_OUTGOING_CALLS
android.permission.WRITE_SETTINGS
android.permission.WRITE_SECURE_SETTINGS

(2-1)隐私级

拥有该级别权限的应用可以直接访问或通过函数获得用户的隐私信息。该级别的权限若单独符合破坏级条件,则隶属于破坏级而非隐私级。这一类权限包括:

android.permission.ACCESS_FINE_LOCATION
android.permission.ACCESS_COARSE_LOCATION
android.permission.READ_CALENDAR
android.permission.READ_CALL_LOG
android.permission.READ_CONTACTS
android.permission.READ_PHONE_STATE
android.permission.RECORD_AUDIO
android.permission.RECEIVE_SMS
android.permission.READ_SMS
android.permission.WRITE_SMS
android.permission.RECEIVE_MMS

(2-2)通信级

拥有该级别权限的应用可以与系统外的部分进行通信。这一级别的权限与隐私级的权限同时存在,可造成用户隐私信息泄露。同样,该级别的权限若单独符合破坏级条件,则隶属于破坏级而非通信级。这一类权限包括:

android.permission.INTERNET
android.permission.SEND_SMS
android.permission.WRITE_EXTERNAL_STORAGE

(3)其他

这里指不在以上范围内的权限,通常属于 normal 权限,应用申请可以直接授权,或应用的自定义权限。对这类权限,检测系统不考虑其安全隐患。

至此,用户交互部分的检测系统组件基本完成。将预处理部分的逆向与特征提取模块、最终分类器结果、权限

组合判断模块与用户交互界面结合,即得到检测系统的用户交互部分。除最终分类器结果可能进行更新,需要作为资源文件存在外,其余部分可以打包成为一个整体。

6 展望

对于下一步的工作内容,应当包括以下几点。

(1)恶意代码段判定模块的添加

本文所给出的检测系统不依赖于特定的恶意代码段,但也正因如此,该系统对于已知的恶意代码家族不能达到传统方法检测的高准确率。在现有的检测系统中加入记录恶意代码段的数据库,或增加自学习模式,都可以有效地解决这一问题。

(2)检测系统需要建立配套的运行维护机制

当前系统仅是建立在少量恶意样本下的原型系统,若需要扩大规模并推广,则必须要解决两个问题:一是对于正常样本与恶意样本的搜集,需要持续地工作并保证分类器的及时更新,该检测系统才能在恶意软件不断变化发展的环境下生存下来;二是当前的版本依然只能在实验 PC 上运行,如需推广,则必须建立云端服务器与服务器—客户端通信机制,建立在线检测平台与检测软件以满足用户需求。

7 结束语

为了适应当前 Android 平台的软件环境,应用自行给出的基于定损曲线面积的分类器评价标准,通过多种组合算法横向对比的方式选出了相对检测精度最高的分类器结果。以该分类器结果为核心,实现了一个模块化、准确率高、运算速度快、资源消耗低的 Android 恶意软件检测系统原型,拥有一定的实用价值。但该系统仍然有不适合处理加密软件、受限于样本规模等静态检测系统普遍存在的问题,需要在以后的工作中加以解决。

参考文献:

- [1] Alcatel-Lucent. Motive security labs malware reports-H1 2015 [EB/OL]. (2015-09-20)[2016-07-30]. <http://resources.alcatel-lucent.com/asset/189669>.
- [2] SHABTAI A. Malware detection on mobile devices[C]//The 11th International Conference on Mobile Data Management, May 23-26, 2010, Kansas City, Missouri, USA. New Jersey: IEEE Press, 2010: 289-290.
- [3] 王志国,侯银涛,石荣刚. Android 智能手机系统的文件实时监控技术[J]. 计算机安全, 2009(12): 42-44.

- WANG Z G, HOU Y T, SHI R G. The file real-time monitoring technology based on the android smart phone system [J]. Computer Security, 2009(12): 42-44.
- [4] ENCK W, GILBERT P, CHUN B. TaintDroid: an information-flow tracking system for real-time privacy monitoring on smartphones[C]//9th USENIX Symposium on Operating Systems Design and Implementation, October 4-6, 2010, Vancouver, BC, Canada. New Jersey: IEEE Press, 2010.
- [5] 冯博, 戴航, 慕德俊. Android 恶意软件检测方法研究 [J]. 计算机技术与发展, 2014, 24(2): 149-152.
- FENG B, DAI H, MU D J. A research of malware detection for android[J]. Computer Technology and Development, 2014, 24(2): 149-152.
- [6] 卿斯汉. Android 安全研究进展 [J]. 软件学报, 2016, 27(1): 45-71.
- QING S H. Research progress on Android security[J]. Journal of Software, 2016, 27(1): 45-71.
- [7] DAVIES S, RUSSELL S. Np-completeness of searches for smallest possible feature sets [C]//AAAI Fall 94 Symposium on Relevance, July 31-August 4, 1994, Seattle, Washington, USA. New Jersey: IEEE Press, 1994: 37-39.
- [8] 刘华文. 基于信息熵的特征选择算法研究 [D]. 长春: 吉林大学, 2010: 21-28.
- LIU H W. A study on feature selection algorithms using information entropy[D]. Changchun: Jilin University, 2010: 21-28.
- [9] SCHMIDT A, BYE R, SCHMIDT H, et al. Static analysis of executables for collaborative malware detection on android[C]//The 9th IEEE International Conference on Communications, June 14-18, 2009, Dresden, Germany. New Jersey: IEEE Press, 2009: 1-5.
- [10] 童振飞, 杨庚. Android 平台恶意软件的静态行为检测 [J]. 江苏通信: 技术与实践, 2011: 39-47
- TONG Z F, YANG G. Static behavior detection of malware on android platform [J]. Jiangsu Communication: Technology Practice, 2011: 39-47
- [11] 欧阳博宇, 刘新, 徐婵, 等. 基于支持向量机的恶意软件行为评估系统[J]. 计算机应用, 2015, 35(4): 972-976, 980.
- OUYANG B Y, LIU X, XU C, et al. Malware behavior assessment system based on support vector machine [J]. Journal of Computer Applications, 2015, 35(4): 972-976, 980.
- [12] 计智伟, 胡珉, 尹建新. 特征选择算法综述 [J]. 电子设计工程, 2011, 19(9): 46-51.
- JI Z W, HU M, YIN J X. A survey of feature selection algorithm[J]. Electronic Design Engineering, 2011, 19(9): 46-51.
- [13] HUANG C, YANG D, CHUANG Y. Application of wrapper approach and composite classifier to the stock trend prediction[J]. Expert Systems with Application, 2008, 4(34): 2870-2878.
- [14] GALAR M, FERNANDEZ A, BARRENECHEA E, et al. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches [J]. IEEE Transactions on Systems Man & Cybernetics Part C, 2012, 42(4): 463-484.
- [15] 中国互联网协会. 恶意软件定义细则[EB/OL]. (2007-06-15)[2016-07-01]. <http://www.isc.org.cn/hdzt/feyrj/listinfo-4196.html>. Internet Society of China The definition of malicious software[EB/OL]. (2007-06-15)[2016-07-01]. <http://www.isc.org.cn/hdzt/feyrj/listinfo-4196.html>.
- [16] CLAUD X. 移动安全这五年 [EB/OL]. (2014-10-01)[2016-07-10]. <http://www.csdn.net/article/2014-10-01/2821943-safe-5-years>. XIAO C. Mobile security in 5 years [EB/OL]. (2014-10-01)[2016-07-10]. <http://www.csdn.net/article/2014-10-01/2821943-safe-5-years>.
- [17] UNUCHEK R. The most sophisticated android trojan[EB/OL]. (2013-06-06)[2016-07-10]. <http://securelist.com/blog/research/35929/the-most-sophisticated-android-trojan/>.
- [18] GUYON I, ELISSEEFF A. An introduction to variable and feature selection [J]. Journal of Machine Learning Research, 2003(3): 1157-1182.
- [19] ZHANG R. ROC curve-threshold evaluation criteria [EB/OL]. (2012-03-16)[2016-07-30]. <http://blog.csdn.net/abcjennifer/article/details/7359370>.
- [20] 王科欣, 徐辉. 基于最小错误率与最小风险的贝叶斯分类比较与研究[J]. 科技信息, 2009(23): 236, 245.
- WANG K X, XU H. Comparative study on minimum error rate bayes and minimum risk bayes [J]. Science & Technology Information, 2009(23): 236, 245.

[作者简介]



陈昊(1988-),男,北京大学软件与微电子学院硕士生,主要研究方向为通信及网络安全、移动安全、安全架构及设计、安全评估与渗透测试等。



卿斯汉(1939-),男,北京大学软件与微电子学院教授,中国科学院软件研究所首席研究员,主要研究方向为移动安全、可信计算、云安全、操作系统安全等。