第 44 卷 第 2 期 计 算 机 工 程 2018 年 2 月
Vol. 44 No. 2 Computer Engineering February 2018

· 安全技术 · 文章编号: 1000-3428(2018)02-0187-06 文献标志码: A 中图分类号: TP316

安卓 APP 安全加固系统的分析与设计

赵跃华,刘 佳

(江苏大学 计算机科学与通信工程学院,江苏 镇江 212013)

摘 要: 为提高 Android 应用软件的保护能力,以增加攻击者破解难度,同时不影响 APP 运行效率为目标,在 APP 说明文档自动识别的基础上,设计一套 APP 安全加固系统。通过对 APP 说明文档进行关键词的自动识别,进而由关键词确定其 APP 的安全需求,根据安全需求给出对应的安全组合加固方案,实现具体的安全加固。分析结果表明,该系统能以合适的加固方案增加 APP 的安全性,有效保护了用户和开发者的合法利益。

关键词: 安卓系统;文档识别;关键词;应用安全;加固

中文引用格式: 赵跃华,刘 佳. 安卓 APP 安全加固系统的分析与设计[J]. 计算机工程,2018,44(2):187-192.

英文引用格式: ZHAO Yuehua, LIU Jia. Analysis and Design of Android APP Security Reinforcement System[J]. Computer Engineering,2018,44(2):187-192.

Analysis and Design of Android APP Security Reinforcement System

ZHAO Yuehua, LIU Jia

(School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China)

【Abstract】 In order to improve the security of Android application software and increase the difficulty of the attacker without affecting the running efficiency of APP as the target, an APP security reinforcement system is designed on the basis of the automatic identification of APP documents. The system identifies the key words by automatic identification of the APP documents. In addition, it determines the security requirements of APP according to the key words, and then gives the corresponding security combination and reinforcement scheme by the security demands to realize the concrete safety reinforcement. The analysis results show that the system can increase the security of APP with the appropriate reinforcement scheme, and protects the legitimate interests of users and developers effectively.

【Key words】 Android system; document recognition; key word; application security; reinforcement

DOI: 10.3969/j. issn. 1000-3428. 2018. 02. 033

0 概述

随着手机平台技术的不断发展,针对各类手机平台的恶意软件层出不穷,这些恶意软件已经威胁到了手机用户的隐私及财产安全[1]。据中科院最新调研报告显示,当前 68.6% 的安卓手机用户正面临移动安全威胁[2]。

针对当前 Android 手机存在的安全威胁,文献[3]提出了使用代码混淆的方式对 Dalvik 字节码程序进行保护的方法,该方法对代码的静态保护效果好,但不能防御攻击者的动态篡改,其防御功能单一,综合性不足。文献[4]提出一种 Android 平台代码保护方案,该方案对加密算法的依赖性较强,若密钥被泄露,则该保护手段能够被轻松破解。文献[5]提出一种利用 Android 本地代码来产生自修改 Dalvik 字节码的方法,该方法能够使软件的破解难度上升,但要是攻击者直接破解本地代码,那么仍然无法保证软件的安全性。

随着技术的发展,各种加固技术层出不穷,若仅使用一种加固技术难以抵御各种攻击,难以保证 Android 应用的安全性。因此,本文考虑采用多种加固技术组合的方式对 Android 应用进行加固。但是加固技术多种多样,不同 Android 应用的特点和加固需求各不相同,怎样合理地选取加固技术,既能满足 APP 的安全加固需求,又能最大限度保证 APP 的运行效率,这对开发者及使用者都具有较好的实际价值。

1 前期分析

1.1 攻击及安全加固框架

未加固的原始 APP 易遭受逆向、篡改等各种攻击,手机中用户隐私数据(通讯录、个人信息数据、电子邮件等)的泄露问题也更加突出[6]。在对当前

作者简介:赵跃华(1958—),男,教授、博士,主研方向为信息安全;刘 佳,硕士研究生。
收稿日期:2017-01-03 修回日期:2017-03-07 E-mail:wutongyuliujia@foxmail.com

APP 面临的攻击、相应加固技术和文档自动识别技术研究分析的基础上,提出并实现一种自动化的、高效的安卓应用加固系统。该系统能够自动识别 APP 说明文档,确定 APP 加固安全需求,根据安全需求高效地组合加固技术,实现对 APP 的安全加固,攻击及其安全加固示意图如图 1 所示。

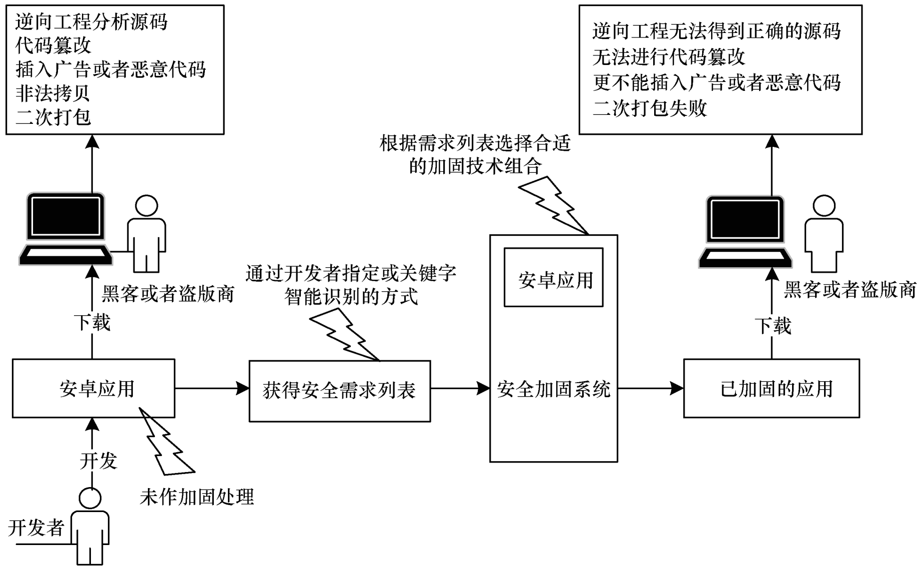

图 1 攻击及安全加固示意图

1.2 攻击及其加固技术分析

针对 APP 遭受的不同攻击,分析其对应的加固技术,如表 1 所示。根据各加固技术的攻击全覆盖、安全性、效率等因素,以加固安全性高且加固后 APP 运行效率高为目标,对加固技术进行筛选、归类,得到能覆盖主要攻击的加固技术最小集,即加壳、代码混淆、软件水印、调试状态监测 4 种加固技术。

表 1 攻击及其对应的加固技术

攻击	加固技术
篡改攻击	加壳,代码完整性校验
静态分析攻击	代码混淆,动态加载
盗版	软件水印技术
逆向攻击	代码混淆
反编译攻击	代码混淆,加壳
动态分析攻击	加壳,调试状态检测法

加壳:将源程序嵌套到壳程序中,这样源程序就很难被非法修改或反编译。加壳技术能有效保护软件版权,防止软件被破解。

代码混淆:通过插入与功能无关的代码和使用晦涩名字重命名类、字段和方法,对代码进行优化和混淆。混淆不影响软件的正常功能,但是使源码的可阅读性大大下降,从而有效保护软件源码。

软件水印:将版权保护信息和身份认证信息隐藏在软件产品中,以鉴别出非法复制和盗用的软件产品,达到有效保护软件产品合法传播的目的。软件水印技术主要特点是秘密性、可复制性、隐藏信息量。

调试状态检测法:该技术原理是检测当前 APP 的调试状态是否开启,是否有进程在监测程序的运行。其主要作用是防止动态分析、动态调试攻击。

1.3 关键词与加固技术对应关系的建立

APP 的说明文档中包含了丰富的信息,其中的关键词能反映出该 APP 的加固需求,如:若有“支付”“购物”等关键词,说明该 APP 与支付交易等业务相关,则该 APP 对安全性要求高,APP 对加固强度的需求也高。本文将关键词作为纽带,连接攻击与对应的加固技术,其关系如图 2 所示。

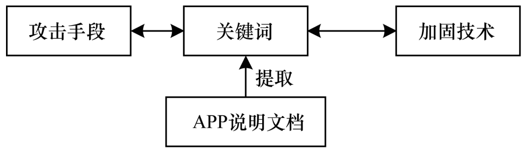

图 2 常见攻击、关键词、加固技术对应关系

研究表明,不同种类的 APP 受到的攻击及强度不同,因此加固需求也存在一定的差别[7]。首先根据 APP 的用途对 APP 进行分类,根据以往经验,可将 APP 分为金融类、社交类、影音类、游戏类、信息类、其他类[8]。

使用具有公信力且加固技术已知的 APP 说明文档,通过人工统计的方法,得到每种类型 APP 的关键词与其所用的加固技术和加固强度的对应表,这也是本文在进行 APP 安全级别判断及加固技术选择时的依据。后期在使用的过程中,也将会根据新的 APP 说明文档不断完善和更新关键词表。

需要设计关键词提取算法,对不同种类的 APP 分别使用该种 APP 的说明文档样本进行训练学习,训练出针对不同 APP 种类的提取模型。在对 APP 进行加固时,首先使用该 APP 的提取模型提取出 APP 说明文档中的关键词,然后查询上述关键词表,最终确定 APP 的加固方案。

2　系统方案

系统方案如图 3 所示,通过提取说明文档中的关键词,与 1.3 节中的关键词表相匹配,进而得出 APP 的加固方案,本文的工作重点是 APP 说明文档的关键词提取。

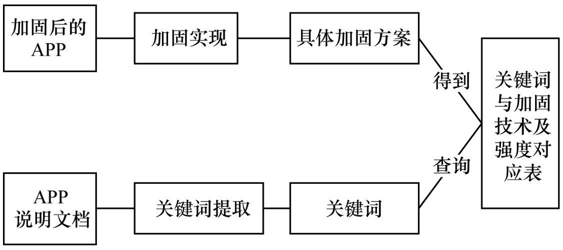

图 3　系统方案图

3　关键词提取算法的设计

对当前流行的关键词提取技术进行了分类研究,最终确定采用候选词权重排序自学习的方法进行关键词提取。首先选取 APP 说明文档,进行分词处理,之后进行关键词提取算法的训练,最终得出关键词提取模型。关键词提取步骤如图 4 所示。

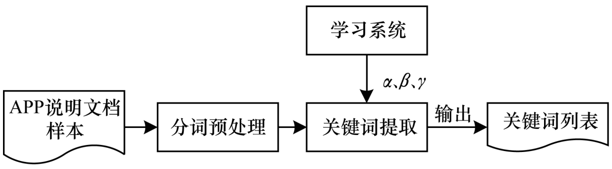

图 4　关键词提取步骤

3.1　分词预处理

经过分词,将 APP 说明文档原始样本转化为关键词提取训练样本,过程如图 5 所示。

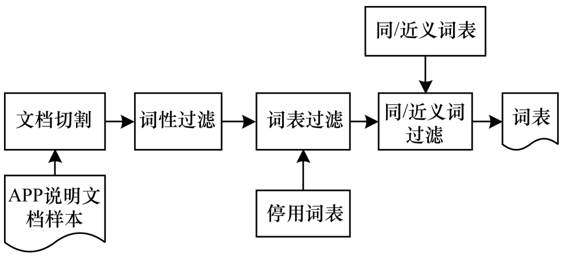

图 5　分词处理过程

3.2　关键词提取

经过分词预处理得到分词后的词语集。本文综合考虑词频 tf_i、区域位置 dis_i、词距 $dis_i = \frac{las_i - fir_i + 1}{sum}$这 3 个特征项,提出了候选词权重计算函数如下:

$$weight_i = a \times tf_i + \beta \times loc_i + \gamma \times dis_i \tag{1}$$

其中,$weight_i$ 表示候选词 sum 的权重,α、β、γ 为 3 个调节因子。

3.2.1　词频

词频特征项 α、β、γ 表示词语在文本中出现的次数,是对词语是否关键的一种常见测度[9]。本文采用公式 $tf_i = \frac{fre_i}{1 + fre_i}$对词频进行计算,$\alpha$、$\beta$、$\gamma$ 为候选词出现的次数。使用非线性函数计算词频基于 2 点考虑:1)词语的词频和 α、β、γ 类成正比,随着词语出现次数的增大逐渐向 1 收敛,即词出现越多,成为关键词的可能性越大,符合词频价值。2)随着词语出现次数的持续增加,词频对词语成为关键词的贡献越来越小,即成为关键词的可能性不会随着次数的持续增加而无限增长,这与语言实际情况相符。

3.2.2　区域位置

经研究发现[10],词语出现在首段中比出现在其他段落更能反映文档主题,摘要中的词语比正文中的词语对文档主题的贡献大,而标题中的词语对文献主题的贡献最大。因此,对于区域位置 loc_i 的计算,设 w_i 为根据分词位置标记的位置值,本文将文本位置区别为 3 种情况:标题中 w_i 设置为 10.0;摘要或首个段落 w_i 设为 5.0;其他段落 w_i 设置为 3.0。若词语在各个位置都出现,则取其最高值。在分词处理时,就可标记词语的位置值,然后使用规约化公式式(2)进行处理,归约化是使数值在(0,1]范围内,能够消除指标之间的量纲影响,解决数据指标之间的可比性。位置值 w_i 经过数据归约化处理后,该指标处于同一数量级,适合进行综合对比评价。

$$loc_i = \frac{w_i - 1}{w_i + 1} \tag{2}$$

3.2.3　词距

词距 dis_i 是指文本中词语首次出现的位置和最后一次出现的位置之间的距离。随着文本长度的增加,局部关键词可能影响文本整体关键词。因此,为了滤除局部关键词,将词距引入关键词的抽取算法中。根据以往的实验分析,采用如下公式计算候选词的词距[11]:

$$dis_i = \frac{las_i - fir_i + 1}{sum} \tag{3}$$

其中,fir_i 为该分词第一次出现的位置序号,las_i 为该分词最后一次出现的位置序号,sum 为经分词后的文本中候选词的总数。

3.2.4　调节因子的训练

在确定了计算 3 个特征项值的公式后,需要考虑如何确定调节因子 α、β、γ,使其更加合理地反映各个因素对权重的贡献程度。对每种类型的 APP 分别进行训练,能得出适合本类 APP 的调节因子。本研究采用机器学习最小均方误差(Minimum Mean

Square Error,LMS)训练法训练公式的调节因子[12],确定调节因子的学习过程如图 6 所示。学习完成后,便得到合适的调节因子 α、β、γ。

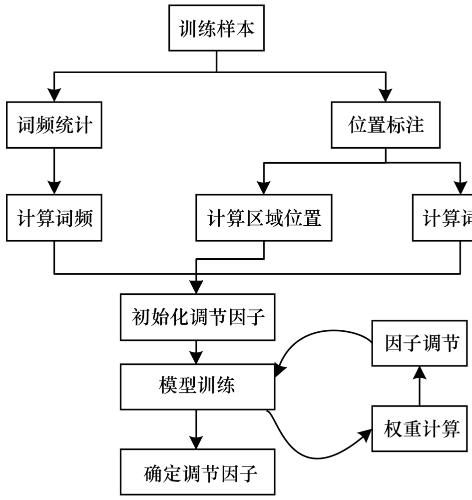

图 6 确定调节因子的学习过程

3.2.5 关键词列表输出

确定了 α、β、γ 后,使用式(1)进行计算而得出候选词的权重,然后根据关键词权重大小对词语进行排序,根据文档的具体需要选取其中的前若干个词语作为文档的关键词即可。

3.3 加固方案的确定

当有新的 APP 需要进行加固时,首先确定 APP 的种类,用 3.2 节中的关键词提取方法进行关键词提取。之后查询关键词集,得到关键词与其对应的加固技术强度。

最后分别得到每个关键词对应的加固技术强度。对于同种技术有 2 个及以上的关键词指向时,取其最大值。$F(x)$的值与加固技术的选择具有直接关系,将加固强度分为 3 个等级(该强度等级可以根据 APP 类型的特点灵活调整)。加固强度与加固策略的对应关系如表 2 所示。

表 2 关联度与加固策略

x 技术的加固强度	x 技术加固策略
$F(x)=0$	不使用 x 技术
$F(x)=1$	使用 x 技术的低强度实现
$F(x)=2$	使用 x 技术的低高度实现

4 加固的实现

由加固技术介绍可知,本文所用的 4 种技术,基本覆盖了所有的攻击,并且互相独立。在加固实现时,首先解压 APK,然后分别运用不同的加固技术进行加固,最后重打包成 APK。下面介绍 4 种技术的实现方法。

调试状态检测法通过在源程序中嵌入检测代码的方法实现,该代码会在 APP 启动时对 proc/xxx/status 等进行检测,预设情况下 status 中 TracerPid 值为 0,若不为 0,说明程序正处于被调试状态,则立即终止程序运行。

对于加壳技术本文采用代码碎片化加壳的方法实现[13],框架图如图 7 所示。该技术的实现主要分 APK 预处理阶段和运行时处理阶段 2 个阶段。在 APK 预处理阶段,主要是提取资源和处理 dex 文件。首先,按照 dex 文件的格式将其分为 8 个独立的片断,并用 AES 算法且使用不同的密钥对片断分别进行加密。而运行时处理阶段完成加载工作和上下文切换工作。首先,会将 dex 文件片分别解密后映射入内存中,然后,为了确保正确的解释并执行碎片化的 dex 文件,需要做一些修复工作,同时,资源的拷贝工作和 Manifest 配置工作也在此完成。接着,在分片运行中,当程序跨片时执行上下文切换工作,从而完整地运行整个 dex 文件。

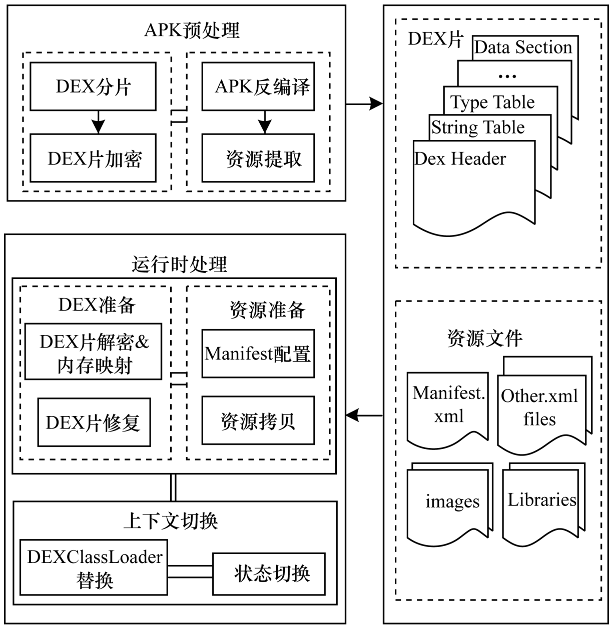

图 7 碎片化加壳实现框架

通过加壳让被保护 dex 文件在其整个生命周期中,始终以碎片化形式存在于进程内存中,这就大大降低了完整的 dex 文件被攻击者窃取的可能性。

代码混淆,通常通过打乱程序中数据的组织形式,变换程序中的数据信息,对函数名和变量名进行变换,尽可能地破坏见名知义的软件工程设计原则等手段来保护源代码,但不会影响程序的功能。本文采用 DEX 混淆重组的方式实现混淆[14],实现过程如图 8 所示。

软件水印技术,是基于大数分解难理论,对软件加水印是将 2 个大数的乘积嵌入 APP,不会涉及对软件其他地方的修改,故将其放在最后一步。本文采用基于动态图的软件水印技术实现水印[15]。

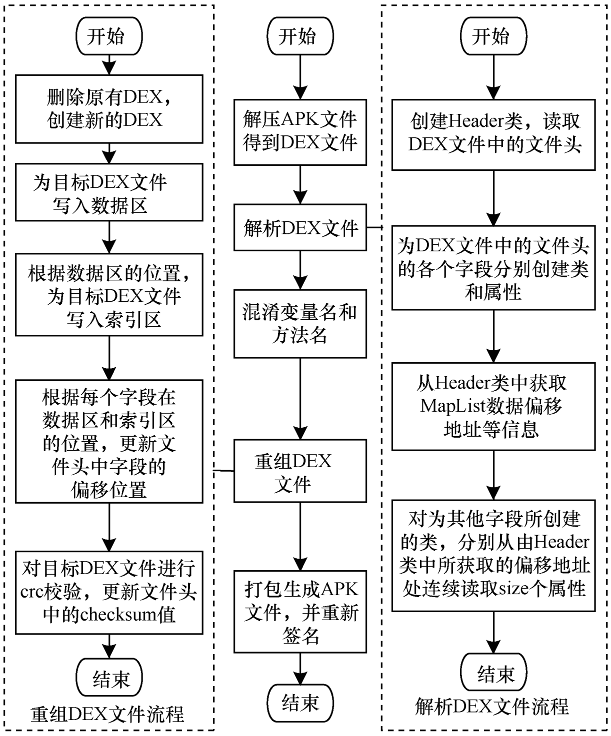

图 8　混淆算法实现流程

5　验证测试

运用本系统对 APP 加固是为了保证 APP 的安全性,但又不能对其本身的功能和效率造成影响,即要保证加固前后功能流程完全一致。运用本系统以金融类 APP 为例,从关键词提取准确性、安全性和性能三方面对方案效果进行分析。

5.1　金融类 APP 关键词与加固技术对应表的建立

用具有公信力且加固技术已知的 APP 说明文档,通过人工统计,得到关键词与其所用的加固技术的关联关系,这也是本文在进行 APP 安全级别判断及加固技术选择时的依据。

由于统计出的关键词数量较多,表 3 给出的是金融类 APP 部分关键词与其对应的加固技术的示例。其中,0 表示该关键词与此加固技术不存在关联关系,1 表示该关键词与此加固技术存在弱关联关系,3 表示该关键词与此加固技术存在强关联关系。

表 3　金融类 APP 部分关键词与其对应的加固技术关系

关键词	加壳	代码完整性校验	代码混淆	软件水印技术
渲染	0	1	1	0
商品	1	0	0	1
…	…	…	…	…
定位	1	0	1	0
破解	0	0	2	1
加密	2	0	1	0

5.2　关键词提取方法的验证

以金融类 APP 为例进行说明,其他种类的 APP 与此类似。选用 60 份金融类 APP 的说明文档,选取其中 50 个说明文档作为训练样本,选取其中 10 个说明文档作为测试样本。使用训练样本,通过机器学习最小均方误差(Least Mean Square,LMS)训练法训练式(1)中的调节因子,然后使用测试样本来验证本关键词提取方法的正确性。

将训练样本输入,通过学习系统训练出一组适合金融类 APP 的调节因子,如表 4 所示。

表 4　金融类 APP 的调节因子

样本数	α	β	γ
60	0.712	0.626	0.667

然后将测试样本输入,使用上述调节因子,提取出关键词。并将提取出的关键词与其文档已知的关键词进行比较,得出提取准确率,如表 5 所示,平均准确率为 84% 。由于说明文档篇幅过短,“值得买”APP 的识别准确率偏低。

表 5　关键词提取准确率

APP 名称	识别出的关键词	已知关键词	准确率/%
51 信用卡管家	信用卡,理财,支付宝,借款,还款	信用卡,理财,支付宝,银行,借款	80
大智慧	投资,股票,足球,资金,理财	炒股,投资,股票,理财,资金	100
借你花	贷款,借钱,资金,在线,呼叫	贷款,资金,借钱	100
钱内助理财	理财,金融,投资,股票,平台	理财,金融,投资,担保,股票	80
值得买	频道,购物,商城,优化,最新	网购,商城,优惠,商品,促销	40
手机贷款	贷款,银行,手机,借款,理财	贷款,手机,借款,银行,理财	100
铜板街手机理财	铜板,资金,理财,银行,手机	手机,理财,理财,银行,资金	80
网银钱包	网银,钱包,金融,理财,密码	网银,钱包,金融,理财,信贷	80
微享猫	财富,赚钱,夺宝,收益,购买	财富,赚钱,夺宝,奖品,收益	80
信用付	贷款,信用,理财,分期,隐私	贷款,信用,隐私,理财,分期	100

关键词提取完后,可通过查询金融类 APP 关键词与其对应的加固技术关系表,获得每个 APP 的加固技术方案。通过本系统得出的加固方案与 APP 已知的加固方案比较而得基本相符,这也进一步验证了本文方案的可行性。

5.3 性能分析

由于关键词提取操作是一个预处理过程，因此，只需分析 APP 加固前后功能和运行效率的变化。运用本文设计的系统对 10 个 APP 样本进行加固，在魅蓝 3s 真机上进行测试，对比加固前后 APP 的大小、启动时间和运行正确性，对照表如表 6 所示。APP 大小通过直接查询的方式获得，APP 启动所需的时间通过 adb logcat 抓取 APP 启动时的 log 而获取。加固后 APP 平均增加 5.07%，启动时间增加 32.78%。

表 6 加固前后系统性对照

APP 名称	大小			启动时间			运行是否正确
	加固前/KB	加固后/KB	增加比例/%	加固前/KB	加固后/KB	增加比例/%	
51 信用卡	16 254	16 952	4.20	1 089	1 458	33.8	是
大智慧	9 854	10 552	7.08	985	1 196	21.4	是
借你花	2 648	2 845	7.43	641	869	35.6	是
钱内助理财	8 495	8 911	4.90	886	1 278	44.2	是
值得买	11 203	11 623	3.75	1 152	1 541	33.8	是
手机贷款	14 527	14 923	2.72	1 296	1 845	42.3	是
铜板街手机理财	5 896	6 262	6.20	758	1 068	40.9	是
网银钱包	9 562	10 412	8.89	994	1 439	44.7	是
微享猫	7 458	7 721	3.50	832	1 091	31.1	是
信用付	22 563	23 016	2.00	1 423	1 836	29.0	是

由表 6 中数据可以看出，在保证 APP 正确运行的情况下，加固后 APP 的大小增加在 5% 左右，这是 Android 系统可以接受的范围；启动时间平均增加了 32% 左右，这对用户的使用体验来说，影响不大。

6 结束语

本文设计了一种加固系统，在保证 APP 运行效率的基础上，高效地实现了 APP 的安全加固。本文方法能够较准确地把握 APP 的加固需求，综合了各种技术的优点，相对于使用单一的技术来说，更加灵活和安全。APP 加固后能够抵御静态分析、反编译等各种攻击，对软件作者及用户来说都具有较好的实际价值。未来可将词语的均匀度也作为一个量化指标用在关键词提取中，这样能够进一步提高关键词提取的准确性，进而更加精确地确定 APP 的加固需求。

参考文献

[1] 刘潇逸，崔 翔，郑东华，等. 一种基于 Android 系统的手机僵尸网络[J]. 计算机工程，2011，37(22)：1-4.
[2] 鲍可进，彭 钊. 一种扩展的 Android 应用权限管理模型[J]. 计算机工程，2012，38(18)：57-64.
[3] KOVACHEVA A. Efficient Code Obfuscation for Android[C]//Proceedings of International Conference on Advances in Information Technology. Berlin，Germany：Springer，2013：104-119.
[4] 张 鹏. 增强本地代码安全性的 Android 软件保护方法[J]. 北京邮电大学学报，2015，38(1)：21-25.
[5] SCHULZ P. Android Security Analysis Challenge Tampering Dalvik Byte Code During Run-time[EB/OL]. (2013-04-24). https://goo.gl/07QBou.
[6] 杨广亮，龚晓锐，姚 刚，等. 一个面向 Android 的隐私泄露检测系统[J]. 计算机工程，2012，38(23)：2-6.
[7] 张 震，张 龙. Android 平台的 Native 层加固技术研究与实现[J]. 计算机与现代化，2016(10)：88-91.
[8] 兰 焱. 基于应用分类的 Android 隐私保护模型的设计与实现[D]. 西安：西安电子科技大学，2014.
[9] 管瑞霞. TFLD：一种中文文本关键词自动提取方法[J]. 机电工程，2010，27(9)：123-127.
[10] 谢 晋. 基于词语跨度的中文文本关键词提取及在文本分类中的应用[D]. 杭州：浙江工业大学，2011.
[11] 丁卓冶. 面向主题的关键词抽取方法研究[D]. 上海：复旦大学，2013.
[12] MEHER P K，SANG Y P. Critical-path Analysis and Low-complexity Implementation of the LMS Adaptive Algorithm[J]. IEEE Transactions on Circuits and Systems，2014，61(7)：778-788.
[13] 樊如霞，房鼎益，汤战勇，等. 一种代码碎片化的 Android 应用程序防二次打包方法[J]. 小型微型计算机系统，2016，37(9)：1970-1972.
[14] 文伟平，张 汉，曹向磊. 基于 Android 可执行文件重组的混淆方案的设计与实现[J]. 信息网络安全，2016(5)：71-77.
[15] 雷 敏，杨 榆，胡若翔，等. 基于动态图的软件水印算法[J]. 成都信息工程大学学报，2015，30(2)：173-175.

编辑 顾逸斐