Inline Text Wrapping Picture

北京邮电大学

硕士研究生学位论文开题报告

学 号: 2019111588

姓 名: 胡杨晔

学 院: 网络空间安全学院

专业(领域): 网络空间安全

研究方向: 应用安全

导师姓名: 张华

攻 读 学 位: 工学硕士

2020年12月11日

|  |  |  |  |
| --- | --- | --- | --- |
| 论文题目 |  | | |
| 选题来源 |  | 论文类型 |  |
| 开题日期 | 2020-12-10 | 开题地点 |  |
| **一、立题依据（包括研究目的、意义、国内外研究现状和发展趋势，需结合科学研究发展趋势来论述科学意义；或结合国民经济和社会发展中迫切需要解决的关键科技问题来论述其应用前景。附主要参考文献目录）（不少于800字）**  研究目的：  本课题旨在研究Android应用的重打包检测，通过研究应用间相似度来确定应用是否为重打包应用，为日常人们在第三方应用商店下载应用时，提供有效的应用安全信息，保障用户安全使用应用。  研究意义：  Android已成为全球最受欢迎的OS平台，并在2020年以85.4％的份额占领了市场，已经成为全球最大的移动操作平台。在OWASP公布的十大移动威胁中，应用重打包排名第九。通过重打包受欢迎的安卓应用来分发恶意软件已经成为恶意攻击者的主要牟利来源。重打包应用是合法应用未经授权的修改版本。重打包应用开发者通常通过反编译APK文件获取到应用的源代码，擅自加入安全性未知的代码进去之后，重新打包成APK文件再分发到应用市场上，误导人们下载进而获取利益。在国家网络安全宣传周发布的《2016中国互联网仿冒态势分析报告》U1中指出，仅2016年1月到8月，在16个行业分类中分别选取的15个热门应用的分析当中，发现高达83%的热门手机应用存在重打包，平均每个应用的重打包量达34个，总感染设备量达6790万部。现在的重打包应用己形成一条黑色产业链，通过伪基站发送钓鱼信息，利用钓鱼网站欺骗用户，诱使用户下载安装恶意重打包应用，截取验证码及用户所有敏感信息。用户倘若在使用移动设备时稍不小心，就会遭遇恶意扣费、短信劫持和隐私窃取等恶意行为的危害，给人身财产和隐私安全带来严重威胁。  国内外研究现状：  针对重打包检测，目前主流的解决技术主要有五种：  1.基于操作码的重打包检测技术。这是最早提出的有效检测重打包的应用，生成指纹的速度快，但是完全无法处理受到混淆和加固的应用。  2.基于令牌的重打包检测技术。对混淆技术有一定的抗性特征，并且可以快速生成，可以进行大规模的数据比较。  3.基于图生成的重打包检测技术。图都是由代码特征进一步转化生成，代表了应用的行为特征，粒度小，比较结果精确。但是图的节点数量和图的结构都会对比较算法产生影响，从而导致准确度的降低。  4.基于UI的重打包检测技术。完全不受当前主流混淆技术影响，并且使用静态分析时能快速生成对应的布局特征。静态检测只能找到显示调用的UI过度关系，使用模板或工具直接生成的APP很容易造成误判。  5.基于资源的重打包检测技术。该技术是静态分析情况下，速度最快的分析模式，但是资源特征非常容易被绕过。  主要文献：  [1]Suarez-Tangil G , Stringhini G . Eight Years of Rider Measurement in the Android Malware Ecosystem[J]. IEEE Transactions on Dependable and Secure Computing, 2020, PP(99):1-1.  [2]Zhan X , Zhang T , Tang Y . A Comparative Study of Android Repackaged Apps Detection Techniques[C]// 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 2019.  [3]Wang H , Guo Y , Ma Z , et al. WuKong: a scalable and accurate two-phase approach to Android app clone detection[C]// International Symposium on Software Testing & Analysis. ACM, 2015.  [4]Suarez-Tangil G , Stringhini G . Eight Years of Rider Measurement in the Android Malware Ecosystem[J]. IEEE Transactions on Dependable and Secure Computing, 2020, PP(99):1-1.  [5] Li L , Bissyande T F , Klein J . Rebooting Research on Detecting Repackaged Android Apps: Literature Review and Benchmark[J]. IEEE Transactions on Software Engineering, 2019:1-1.  [6]Ma J . RepDroid: An Automated Tool for Android Application Repackaging Detection[C]// 2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC). IEEE, 2017.  [7]Yue S , Sun Q , Ma J , et al. RegionDroid: A Tool for Detecting Android Application Repackaging Based on Runtime UI Region Features[C]// 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE Computer Society, 2018.  [8]Luman Shi, Jiang Ming, Jianming Fu, Guojun Peng, Dongpeng Xu, Kun Gao, and Xuanchen Pan. 2020. VAHunt: Warding Off New Repackaged Android Malware in App-Virtualization's Clothing. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20Association for Computing Machinery, New York, NY, USA, 535–549.  [9]Khanmohammadi K, Ebrahimi N, Hamou-Lhadj A, et al. Empirical study of android repackaged applications[J]. Empirical Software Engineering, 2019, 24(6): 3587-3629.  [10] Lin M, Zhang D, Su X, et al. Effective and scalable repackaged application detection based on user interface[C]//2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2017: 1-6.  [11]黎桐辛,韩心慧,简容,肖建国.支持加壳应用的Android非侵入式重打包方法研究[J].北京大学学报(自然科学版),2018,54(06):1147-1158.  [12]汪润,王丽娜,唐奔宵,赵磊.SPRD:基于应用UI和程序依赖图的Android重打包应用快速检测方法[J].通信学报,2018,39(03):159-171.  [13]汪润,唐奔宵,王丽娜.DroidFAR:一种基于程序语义的Android重打包应用抗混淆检测方法[J].武汉大学学报(理学版),2018,64(05):407-414. | | | |

|  |
| --- |
| **二、研究内容和目标（说明课题的具体研究内容，研究目标和效果，以及拟解决的关键科学问题。此部分为重点阐述内容）（不少于2500字）**  拟解决的关键科学问题：  重打包应用检测本质上就是检测应用之间的相似性，当两个应用之间的相似性超过了一定的阈值，就判断这对应用是一对重打包应用对。拟解决的关键科学问题主要分为以下三部分：  1.如何面对未知的数据集时能划分一个合适的阈值，使其能够有效判断应用是否是重打包应用。当前所有的重打包检测的研究中，都是通过相似度是否超过指定阈值，判断应用是否是重打包。但是对于大规模的相似度比较，选择一个合适的阈值是非常困难的。当前的所有研究中，阈值的划分都是通过测试大量数据集，选取一个在所有数据集中产生漏报和误报相对最小的阈值。这种阈值的选取过程通常是漫长且复杂的，并且由于真实应用市场中重打包应用的数据分布是不可预知的，根据存在的数据集设置的固定阈值很可能产生概念漂移。即应用市场中的应用对相似度数据的统计特性可能随着时间的推移以不可预见的方法发生变化，从而导致漏报和误报的发生。  2.如何选取合适的特征，快速过滤出可疑的重打包应用。因为使用混淆和加固技术的规模与强度的不同，基于代码特征的静态检测技术对于良性应用是有效的，对重打包应用是无效的。Suarez-Tangil等人在2020年发表在IEEE Transactions on Dependable and Secure Computing的论文《Eight Years of Rider Measurement in the Android Malware Ecosystem》中指出在2017年中谷歌应用市场只有少于25%的应用使用了混淆技术，但是超过90%的重打包应用使用了高级混淆技术。这说明对于静态检测中结果最精确的基于代码特征的技术而言，分析重打包应用在时间上是昂贵的和不准确的。因为当前的混淆和加固技术并没有对资源文件进行加密，所以可以选择资源文件的特征，对重打包应用进行判断，但是由于大多数资源文件并没有与代码有关联，删除、增加或者修改资源都不会对应用产生任何影响，因此选择的特征需要与代码有关联，删除或修改这些特征代表的资源文件时，会对应用正常运行造成影响。  3.如何为混淆或加固应用生成CG。调用图（Call Graph），也被称为函数调用图，通常简写为CG。CG是静态检测中生成的应用特征，反映了应用各函数和API之间的调用关系。通过分析API在各个函数之间的调用关系，可以预测应用在运行时产生的所有行为。但是由于第二点中提到的混淆和加固技术的影响，静态检测可能并不能生成准确的CG。为了逃避可能存在的安全检测，很多重打包技术都会使用混淆或加固技术，并且在应用中使用动态加载技术来进一步逃避安全检测。动态加载的代码只有在程序运行时才会被加载，静态检测完全无法感知到动态加载的代码。这样被隐藏的代码只有在动态运行中，才能被应用执行从而进一步被安全系统检测。但是动态检测的局限性，导致在动态检测中只能获取当前应用正在运行方法的方法名，从而获取一段基于时间的方法调用序列，这样的序列并不具有方法与方法间的调用关系，因此不能直接使用这些信息生成CG。  研究内容：  1.基于相似度分数数据动态生成阈值。为了解决问题1，需要设计一种合适的动态划分阈值方案以判断应用是否为重打包应用。首先，需要通过大批量的测试计算出应用相似度数据，通过针对相似度数据进行分析，提取出相似度数据的分布特征，根据分析得出的结果，设计相对应的动态阈值计算算法。  2.基于静态特征快速过滤出可疑应用对。由于重打包应用绝大多数都使用了高级混淆应用，采用代码特征快速过滤出可疑应用对是不可行的，因为这将消耗大量的时间资源，并且生成的特征很可能受到混淆和加固技术影响导致特征不准确。所以首先需要研究分析哪些应用特征不被混淆和加固技术影响。然后进一步研究这些特征的生成时间和对重打包应用的过滤效果。最后，设计对比试验方案，分别测试被混淆和加固技术影响的特征和不被影响的特征对重打包应用的过滤效果和特征消耗时间，选取出最合适的特征。  3.基于动态检测结果生成CG。寻找Android系统内部是否存在能够调试或追踪当前运行方法状态的API。通过动态Hook正在运行的方法，向其注入相对应的API，达到动态获取混淆或加固应用的函数调用数据的目的。再通过进一步分析应用相对应的函数调用数据，设计拟合算法，使用数据拟合拼接生成CG。  研究目标：  目标是设计一个能够快速且准确检测重打包应用的系统，并且该系统能够根据当前的相似度数据分布情况，动态调整阈值，能够极大的避免未知数据带来的概念漂移。  现有的应用检测多是使用基于代码特征的静态检测技术对重打包应用进行检测，但是由于混淆、加固技术的普及，重打包应用往往会使用这些技术去逃避基于代码特征静态检测。由于代码特征的局限性，目前也有很多应用检测是根据资源特征进行检测的，但是由于资源特征并不能代表应用的行为特征，并且资源特征通常都是易于修改的，单纯的使用资源特征进行检测会造成很大的误报和漏报。  对于用户来说，重打包应用的威胁是极大的。首先，由于重打包应用通常都是针对受欢迎的良性应用，在图标，界面等用户能够感知的特征上，重打包应用和原始应用并没有任何区别。因此，用户通常无法分辨重打包应用和原始应用有哪些区别，并且由于原始应用的知名度，用户已经默认了重打包应用是安全的。这也就导致了用户完全不能意识到自己触发了恶意行为或观看了重打包应用恶意嵌入的广告。  针对于个人手机在各种应用商场下载的某些应用，这些应用可能没有被应用市场检测，存在各种各样的安全问题，甚至在官方应用商店下载的某些应用也因为审查不力而存在着应用重打包的问题。并没有一个平台能够提供一个检测系统，能够对于手机用户提供一个重打包检测服务。所以如果能提供一个检测系统，它可以快速准确的判断出当前应用是一个重打包应用，存在安全风险，提醒用户需要谨慎安装，这将降低用户被恶意应用攻击的风险。  效果：  对于大多数人来讲，他们可能没有意识到重打包应用的存在。重打包应用有着和正常应用一样的图标、功能和界面，没有明显的能被用户感知的恶意行为。用户即使在遇到相同或相似的应用时，也不会对应用的发布者进行仔细判断，而是随便选择一个应用进行安装。对于用户来说，重打包应用和其他正常应用没有任何区别。如果有一个检测系统可以在应用市场自动发现服务器中存在的重打包应用，并且提供给用户检测重打包应用的服务。对于用户来说，服务器端的自动检测能降低用户被重打包应用攻击的风险，提供给用户的重打包应用检测服务也可以让用户主动进行应用检测，能提高用户的安全意识，进一步降低应用的安全风险。 |

|  |
| --- |
| **三、研究方案设计及可行性分析（包括：研究方法，技术路线，理论分析、计算、实验方法和步骤及其可行性等）（不少于800字）**  研究方法：  方法将对应拟解决的科学问题分为三部分。  1.如何面对未知的数据集时能划分一个合适的阈值，使其能够有效判断应用是否是重打包应用。动态阈值对未知的数据集有着高适应性。设置动态阈值触发机制，只有数据触发了机制，才重新计算新的阈值，动态阈值算法参考了图像分割领域的最大类间方差法，该方法计算迅速，并且生成的阈值对重打包应用有着良好的区分度。  2.如何选取合适的特征，快速过滤出可疑的重打包应用。只使用布局文件，通过布局文件快速生成唯一特征，快速过滤出可疑的重打包应用。  3.如何为混淆或加固应用生成CG。通过向活动的OnCreate和OnDestroy方法注入Android自身的API，动态获取两个方法之间所有线程运行时调用的方法数据。基于这些数据，使用拟合拼接算法生成CG。  可行性分析：  可行性分析方法将对应研究方案分为三部分。  1.动态阈值生成时应该保证快速，并且具有良好的区分度。动态阈值触发机制保证了只有能够对当前数据分布特征产生影响的新数据被输入时，才重新计算阈值，降低了阈值计算的平均时间。相似度分数数据的分布会呈现双峰趋势，并且在数据分布中存在的异常数据数量少。数据分布与图像分割中分割前景与背景的图片像素分布非常相似，因此可以参考图像分割领域的最大类间方差法动态生成阈值。  2.重打包应用和原始应用都具有高度相似的界面以欺骗用户下载使用。因此选取布局文件作为特征快速过滤可疑重打包应用。布局文件与代码紧密相连，删除或者修改布局文件或布局文件中的组件需要修改相对应的代码，这对于应用重打包者的代价是巨大的。在面对应用中增加的冗余布局来说，只需要指定一个约束规则，规定每一个布局最多只允许匹配一个布局的情况下生成最佳布局匹配，就能极大的减小冗余布局文件对相似度判定的影响。  3.重打包应用和原始应用都具有高度相似的界面。在动态检测中，如果事件输入器在对应用进行模拟操作时能够保证在相同或相似的界面下，生成的事件基本相同，那么对于重打包应用和原始应用，他们遍历的界面路径就相同，触发的代码行为也基本一致，这样就可以保证重打包应用和原始应用生成CG图的完整程度基本一致。  技术路线：  1.布局相似度检测  对于应用包含的布局文件进行解析，排除可能插入的冗余布局组件或冗余布局，将剩余的布局文件进行相似度的比较。  2.动态检测  对于过滤出的应用对在沙箱环境中进行模拟输入，通过向方法中注入追踪函数，获取应用运行时的方法调用。  实现方法及步骤：  1.对一定规模的应用进行相似度分析，划分一个合适的初始阈值；  2.将应用按顺序解压，解析布局文件，排除布局文件中可能存在的冗余布局或组件，在特征提取中排除这些文件和组件，然后进行特征提取，并比较相似度，通过动态阈值找出可能存在的重打包应用程序。  3.对所有可疑的重打包应用程序在沙箱环境中模拟执行，通过动态注入追踪函数获取线程间的函数调用信息，使用这些信息拟合生成CG。  4.对生成的CG进行相似度比较，相似度超过动态阈值的应用判断为重打包应用。 |

|  |
| --- |
| **四、本研究课题可能的创新之处（不少于500字）**  1.采用动态阈值判断应用重打包，提高了对未知数据集的适应性，降低了人工确定阈值时的复杂性。在当前所有的应用重打包检测中，都是采用人工定义阈值去判断重打包应用。人工定义阈值需要使用大量的数据反复进行测试以确定一个合适的阈值。由于阈值是固定的，面对未知数据集时很可能产生概念漂移，并且精度受到人工限制，从而导致误报率和漏报率的上升。  2.只使用布局文件作为静态分析特征，快速过滤可疑的重打包检测。当前采用布局文件作为静态分析特征的重打包检测研究中，都是将布局与代码相结合的方式去除应用中可能存在的冗余布局，往往并没有考虑混淆和加固技术对于提取布局与代码的关联特征产生的影响。或者直接采用动态获取运行时使用的布局文件，以此来获取布局与代码的关联特征，花费的时间成本是昂贵的。  3.使用动态检测生成的数据，拟合生成CG图。CG图作为静态分析中能够反映应用行为的特征，通常结合控制流图等特征一并使用，但是在面对动态加载或者加固技术时，由于重打包应用源代码被隐藏，导致CG图生成不全面，从而导致比较应用相似性时产生误差。而动态检测只能直接获取到基于时间的函数调用序列，由于缺少函数的直接调用关系，这些数据并不能直接用来生成CG。 |
| **五、研究基础与工作条件（1.与本项目相关的研究工作积累基础 2.包括已具备的实验条件，尚缺少的实验条件和拟解决途径）（不少于500字）**  1.与本项目相关的研究工作积累基础  这个项目的需求很多，需要多方面的技术，需要掌握的语言有java、python，需要熟悉的框架的有Android框架、Xposed框架，需要学习的技术主要有Android应用逆向分析，树和图的相似度比较，图像分割领域中的动态阈值算法的原理。  当前已经具备的试验条件，目前可以使用大量的开源逆向工具，对应用程序进行逆向解包，提取出应用的布局文件，通过对布局文件的解析得到布局文件中的组件树。动态生成CG可以使用沙箱模拟动态环境，对应用进行仿真运行。动态注入追踪API，然后对应用进行模拟操作，获取应用运行时方法的调用信息，通过调用信息拟合CG。  当前主要缺少对比布局文件的组件树的相似度的方案，CG的相似度快速比较方案。由于重打包应用并没有一个基准的数据集，所以目前还缺少一个已经打上正确标签的重打包应用数据集。  2.拟解决途径  布局文件的组件树可以通过查找文献找到树的相似度比较方案，或者重新从布局文件中提取容易比较相似度的特征。CG相似性比较涉及到图的相似性比较，可以查找最新的关于图比较方向的文献，找到合适的快速比较方法。2019年Li等人在IEEE Transactions on Software Engineering发布的论文《Rebooting Research on Detecting Repackaged Android Apps: Literature Review and Benchmark》中公布了他们使用的基于人工分析和系统分析重打包数据集并在github上开源，目前数据集仍然在不断的更新。可以使用当前公布的数据集进行试验测试。 |

**学位论文工作计划**

|  |  |  |
| --- | --- | --- |
| 时间 | 研究内容 | 预期效果 |
| 2020年11月-2020年12月 | 阅读相关方向论文、选题 | 完成开题报告 |
| 2020年12月-2021年3月 | 通过布局文件过滤可疑重打包应用对 | 可以快速过滤重打包应用对；产出当前数据动态生成合适阈值的算法 |
| 2021年3月-2021年6月 | 分析相似度数据分布，完成动态阈值算法的设计 | 动态阈值在未知数据中判断重打包是有效的；设计实现一个完整的系统，基于系统产出一篇论文 |
| 2020年6月-2020年9月 | 使用动态检测结果拟合生成CG | 动态检测生成CG能够精确判断重打包应用；基于动态检测结果生成CG的检测方法 |
| 2020年9月-2021年10月 | 应用程序框架搭建 | 完成运行并测试所有流程 |
| 2021年10月-2022年1月 | 论文写作 | 论文最终完成 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 评  定  小  组  成  员 | 姓 名 | 职 称 | 导师类型 | 单位名称 | 职务 |
| 秦素娟 | 教授 | 博、硕导 | 北京邮电大学 | 组长 |
| 金正平 | 副教授 | 硕导 | 北京邮电大学 | 成员 |
| 李文敏 | 副教授 | 博、硕导 | 北京邮电大学 | 成员 |
| 时忆杰 | 工程师 |  | 北京邮电大学 | 成员 |
| 张华 | 副教授 | 博、硕导 | 北京邮电大学 | 成员 |
|  |  |  |  |  |
| 导师意见：  同意开题。 | | | | | |
| 导师（签名）：  日期： 年 月 日 | | | | | |
| 开题报告小组意见： | | | | | |
| 组长（签名）：  日期： 年 月 日 | | | | | |
| 学院意见（签章）： | | | | | |
| 负责人：  日期： 年 月 日 | | | | | |