移动终端的多维度隐私泄露评估模型研究

李涛[1)] 王永剑[2)] 邢月秀[1)] 胡爱群[1)]

[1)](东南大学信息科学与工程学院，南京 210096)

[2)](公安部第三研究所，上海 200031)

摘 要 移动终端隐私泄露问题日益严重，现有的单一检测方法存在一定的局限性，本文基于应用程序的架构，提出了一种包括静态分析、动态分析和数据分析的多维度检测框架，使用静态分析的结果为动态执行提供指导，有利于提高覆盖率和准确率，并分别针对 Android 和 iOS 系统平台进行了泄露行为特征抽取的研究。为量化评估提供更加全面的泄露事件数据和抽象特征描述，在评估的过程中引入用户对隐私对象的预期关注度，提出了带有主观性的隐私泄露评估模型，通过对 Android 和 iOS 应用的测试分析表明，本文的检测框架能够对移动终端应用的隐私泄露事件进行准确高效的检测，评估模型能够反映用户的主观预期，有效弥补了单一检测维度的局限性，为隐私泄露的个性化评估提供了基础理论支撑。

关键词 隐私泄露；评估模型；静态分析；动态分析；数据分析；主观预期

中图法分类号 TP309

Research on multi-dimensional privacy leakage evaluation model for mobile terminals

LI Tao[1)] WANG Yong-jian[2)] XING Yue-xiu[1)] HU Ai-qun[1)]

[1)](School of Information Science and Engineering, Southeast University, Nanjing 210096)

[2)](The Third Research Institute of Ministry of Public Security, Shanghai 200031)

Abstract Privacy leakage of mobile terminals becomes a serious problem with the rapidly development of mobile applications. Leakage detection is one of the important methods to protect user's privacy data. The state-of-the-art researches only use isolated static analysis or dynamic analysis technologies. Static analysis owns benefits of fast speed, but be limited to high false positive. Dynamic analysis performance well in accuracy rating, but its testing speed is slow. Based on application composition, a feature of application contains three dimensions which are code, behavior and data. Code and behavior are related to static and dynamic testing separately. Data testing can be accomplished by analyzing data flow. Being different from aforementioned single analysis technology, this paper proposes a multi-dimensional testing framework taking into account of the overall application structure, which contains static analysis, dynamic analysis and data analysis. The framework firstly analyzes applications' static structure and invoking information to find potential invoking paths of sensitive information. The potential paths are used to guide the subsequent dynamic executing. This method not only improves testing efficiency and coverage rate, but also solves the limitations of single dimensional testing method, which can provide more complete leakage event data for privacy leakage assessment model. Under the proposed framework, the privacy leakage testing is also divided into three layers including data acquiring, eigenvectors forming and quantitative evaluating. During the assessment process, the final quantitative

本课题得到国家自然科学基金(61601113)、国家重点基础研究发展计划基金（2103CB338003）、公安部第三研究所开放课题（C15606）资助。李涛，男，1984年生，博士，讲师，主要研究领域为移动终端安全，隐私保护，可信计算.E-mail: lit@seu.edu.cn. 王永剑（**通信作者**），男，1981年生，博士，副研究员，主要研究领域为网络信息安全.E-mail: wangyongjian@stars.org.cn. 邢月秀，女，1991年生，博士生，主要研究领域为隐私保护.E-mail: 1462878956@qq.com. 胡爱群，男，1964年生，博士，教授，主要研究领域为网络与信息安全.E-mail: aqhu@seu.edu.cn

evaluation results are calculated based on the three-dimensional eigenvectors and user's expectations. In order to use the proposed framework in real systems, behavior characteristic events about privacy leakage are abstracted for Android and iOS platforms. The abstracted events provide comprehensive original leakage data for quantitative evaluation. After acquiring original leakage data, a comprehensive quantitative evaluation method is required to process the data from multi-dimension. As each user has different attention on various privacy information, privacy information leakage has subjective property. A privacy leakage evaluation model is proposed by introducing user's subjective expectation. Under the evaluation model, user firstly labels each sensitive objects with an attention level that contains five ranges. Then a testing events parsing algorithm is used to search the concerned objects from whole leakage events. Finally, the normalized evaluation result is calculated, which combines leakage events and user's subjective. The proposed multi-dimensional testing framework is applied to test privacy leakage on Android and iOS applications separately. The testing results show that multi-dimensional testing system is able to do more comprehensive analysis on application's behavior. Additionally, the detection efficiency is also promoted. Furthermore, we choose 30 typical applications from Android and iOS application market separately to analyze the data type of leakage information. As for the privacy assessment value, two assumed users set their attention levels to each data type separately. The attention levels are used to calculate leakage risk for one Android application and one iOS application. The result shows that the proposed evaluation model reflects user's expectation correctly. The evaluation model overcomes limitation of single dimensional testing method and provides basic theory foundation for personalization evaluation about privacy leakage.

Key words privacy leakage, evaluation model, static analysis, dynamic analysis, data analysis, subjective expectation

1 引言

随着移动互联网的迅速发展，越来越多的人使用移动终端上网，移动终端的应用程序数量也迅速增加，尤其是两大主流移动终端操作系统平台 Android 和 iOS，其应用商店的下载量已经突破每年上亿次。海量应用程序在丰富了移动终端功能的同时，也带来了严重的隐私泄露问题，给用户造成巨大的安全隐患。

根据 360 互联网安全中心发布的《2016 年中国手机安全状况报告》[1]：在 Android 平台上，2016 全年累计截获新增恶意程序样本 1403.3 万个，平均每天新增 3.8 万恶意程序样本，用户感染恶意程序 2.53 亿人次，平均每天恶意程序感染量约为 70 万人次。iOS 平台虽然构造了一个相对封闭的运行环境，但安全问题依然存在，其中影响较大的有 2015 年 6 月份发生的 22 万 iCloud 账号密码泄露事件，以及同年 9 月爆出的开发工具 Xcode 被篡改事件，后者导致 300 多款常用的应用程序被注入木马，受影响的用户数达到了上亿。

针对应用程序安全隐患，Google 和 Apple 公司都部署了应用程序安全审查策略，但只针对官方的 Google play 商城和 App Store，并且审查细节也没有公布，并不能保证用户隐私的安全。而且国内众多用户移动终端中的应用来自于第三方应用市场，其中存在了大量的重打包应用，这些应用有着巨大的隐私泄露风险。

面临移动终端日益严重的安全问题，针对应用软件进行隐私泄露检测是有效的方法，但现有的研究工作大多基于一种检测方法开展的，孤立的使用静态检测或者动态检测都无法避免其固有的局限性。例如静态检测无法对混淆加密过的应用程序代码进行直接分析，需要进行解密处理，分析结果依赖于解密的成功率[1]；Android 应用程序编写过程中使用了大量的隐式函数，在静态扫描过程中对函数调用序列的准确性判断有较大影响[2]；动态检测需要模拟应用程序的真实运行环境，检测速度较慢，并且在实际检测过程中往往难以遍历所有逻辑功能，检出率不高[3]。

同时采用多种检测技术能够综合静态分析和

1 360 互联网安全中心. 2016 年中国手机安全状况报告, http://zt.360.cn/1101061855.php?dtid=1101061451&did=490260073, 2017,02,06

动态分析的优点，但在实施过程中只是孤立的使用各个检测技术得到更多的隐私泄露原始数据，缺少有效的综合分析方法以提高检测效率，也没有合理的量化评估方法来处理多个维度的数据，因此目前关于综合检测框架的研究并没有真正开展。缺少综合量化评估方法导致用户面对庞大的专业泄露事件数据不能有效的进行理解和评估，检测结果不能直观的呈现。在实际检测过程中，静态分析可以得到应用的结构及调用信息，这些信息能够为动态分析提供指导，提高检测效率。在评估过程中用户具有主观特性，对不同类型的隐私泄露事件的关注度不同，每个用户根据自身需求对泄露的评价指标也有差异，需要根据用户主观意愿在不同的泄露事件上有所权衡。

基于此，本文综合静态分析、动态分析和数据分析三个维度，在三个维度间建立相关性，提出适用于移动终端的多维度隐私泄露评估模型，其中包含了多维度检测框架和基于用户主观预期的量化评估方法，有效的对 Android 和 iOS 系统应用程序的隐私泄露行为进行分析和评估。

2 相关研究

移动终端隐私泄露检测主要包括 3 种技术：静态分析、动态分析和数据分析。

已有静态分析的工作（Android 平台的 Kirin[4]、ELF 文件符号信息检测[5]、Apex[6]、FlowDroid[7]、ScanDroid[8]、AppIntent[9]和适用于 iOS 平台的 PiOS[10]、idb[11]）可以分为两个类型：1）对程序代码进行特征匹配。例如 Kirin 系统在应用软件安装时进行基于恶意特征库的权限分析和匹配[4]；通过逆向工程抽取静态 ELF 文件符号信息，与恶意特征库匹配以检测隐私泄露行为[5]；APPIntent 对泄露事件的特征进行了描述和检测[9]。2）静态数据流分析。例如根据函数的调用关系刻画数据流向[7][8]，检测潜在的泄露途径；针对 iOS 平台，PiOS 通过数据流分析检测隐私泄露行为[10]。静态检测方法实现简单，但面对混淆处理的代码时检测精度会下降，无法实现精确检测和持续数据流跟踪。

为了解决静态分析的弊端，动态分析技术日益兴起，其中最著名的是针对 Android 平台的 TaintDroid 框架[12]，通过修改的 Android 虚拟机对敏感数据进行污点标记和跟踪，基于该框架实现的检测系统有 AppFence[13]和 DroidBox[1]，文献[14]基于 TaintDroid 提供的污点跟踪功能对恶意应用的动态行为序列进行特征提取形成恶意行为库，设计了有效的匹配算法提高了应用恶意行为的检出率。CRAXDroid[15]基于系统级符号执行平台 S^2E 实现对网络端口发送数据流的动态监测。针对 iOS 平台的动态检测通过挂钩敏感 API 函数实现[16]，例如 PMP[17]和 DiOS[18]，通过挂钩监控应用程序对用户隐私数据的访问行为。MobileAPPScrutinator[19]针对 Android 和 iOS 平台分别研究了关键 API 的挂钩技术，是一个可以兼容两种平台的检测工具。相对于静态分析，动态分析检测速度较慢，并且容易漏检。

数据分析方法能够直接检测到隐私泄露行为，主要通过检测网络数据包来实现，例如 MobileScope[2]通过网络代理的方式对 Android 和 iOS 平台数据包进行分析。Labyrinth[20]在设置代理监控的同时在应用程序端监控用户输入，通过匹配敏感标记来检测隐私数据泄露。为了提高匹配效率，BayesDroid[21]在匹配的过程中引入贝叶斯分类判决方法。在网络接口进行数据分析检测最大的问题在于不能处理 SSL 数据，因此一些研究者提出了新的思路，例如 Zhou 等人[22]提出一种将网络数据包的时间线和应用行为结合的方法发现非用户主动触发的数据发送行为。Aquifer[23]是一种基于策略的检测框架，通过检测网络数据包是否与访问策略相违背来发现隐私泄露事件。Chen 等人[24]将侧信道攻击的思路引入到隐私泄露检测中，对数据包和状态机的流转关系进行建模，通过网络数据包的大小和流程来判断用户隐私数据。这些检测方法与数据包内容无关，无需解析数据包内容，但检测粒度较为粗糙，无法对隐私泄露的数据进行详细分析。

综上所述，目前针对两大主流移动终端系统的隐私泄露研究中，大多都是采用单一的分析方法，在获取泄露事件数据后，也亟需有效、实用的评估方法对泄露风险进行量化。已有的移动终端安全评估研究开始更多的考虑用户的主观期望，例如文献[25]使用三元组来定义安全威胁，其中一个元素为用户对此事件标识的威胁等级。文献[26]通过用户

1 Droidbox - Android Application Sandbox. https://code.google.com/p/droidbox/ 2013,11,16

2 Mobilescope: Acquired by evidon. http://www.evidon.com/mobilescope 2015.08.12

设置的期望矩阵进行定量评估。文献[27]使用 NLP 语言对用户期望进行形式化描述，并与应用实际事件进行对照。Y. Jing 等人[28][29]使用机器学习的方法对用户期望进行评估，以此作为检测的依据。这些研究考虑用户主观意愿的思想值得借鉴，但需要和检测框架结合进行评估模型的设计。基于此，本文融合三种检测方法提出移动终端隐私泄露综合检测模型，并在此模型下进行特征抽取和量化评估的研究。

3 多维度的隐私泄露评估模型

目前隐私泄露检测主要有静态检测和动态检测两种方法，静态检测速度快，但是误报率高；动态检测速度相对较慢，检测出来的事件都是应用实际发生的调用行为，但在检测过程中由于事先并不确定应用是否存在敏感传输以及应用在哪个操作中发生了敏感传输，测试执行过程中只能尽可能完备的进行路径覆盖，在应用程序逻辑结构比较复杂的情况下，这种检测执行方法效率低下，耗时很长。因此可以通过静态分析应用的结构及调用信息解决动态执行的缺陷，将动态分析和静态分析结合，先进行静态分析寻找可能发生敏感信息调用的路径，为随后的动态执行提供指导，提高检测效率和覆盖率。另外从应用程序的构成来看，主要有代码、行为和数据三个维度，代码和行为分别对应静态检测和动态检测，数据检测可以通过对数据流进行分析实现。

基于此，本文提出了三维检测框架，如图 1 所示，在横向上将静态检测、动态检测和数据检测相结合，通过静态分析得到的应用结构信息为动态检测和数据检测提供执行路径指导，有效解决了单一检测方法的局限性。纵向上将隐私泄露检测分为事件获取、特征向量形成、量化评估三个层次，在评估过程中基于用户主观期望对三个维度特征向量进行综合计算，得到最终量化评估结果。

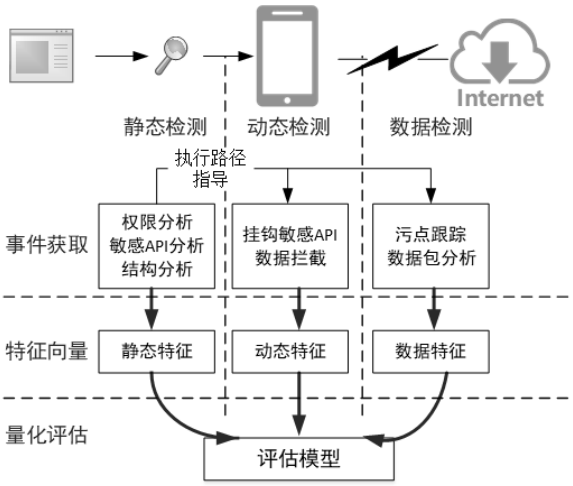

图 1 多维度隐私泄露检测

对应用进行检测的过程中，首先对代码进行静态扫描，分析应用代码中申请的权限和涉及隐私信息的调用，从应用的源代码中抽取用于指导动态执行的静态特征，获取调用关系图，并根据敏感 API 分析结果对包含有调用敏感信息的应用执行逻辑进行重点标记，在随后的动态执行过程中，对于被标记的逻辑组件重点检测，而对于未标记的组件可以快速略过，以此提高检测效率。

静态检测后将应用安装到检测平台中进行动态检测，监控在实际运行过程中的隐私信息调用行为，其中主要牵涉到对敏感 API 挂钩和操作数据拦截。在应用动态运行的过程中对数据流进行监测，对发生的隐私信息发送行为进行记录。

由于 Android 和 iOS 平台的开放性不同，需要针对两个平台进行不同的数据分析技术研究。静态分析、动态分析和数据分析分别对应一个特征向量，最终形成整体泄露事件的特征向量。基于此，将隐私泄露事件定义如下：

定义 1：隐私泄露事件定义为 E，E 包含三个维度，分别是静态特征向量 S、动态特征向量 R 和数据特征向量 D：

$$E = \{S, R, D\} \tag{1}$$

其中，$S = \{s_1, s_2, \ldots, s_n\}$，$n$ 为静态特征向量的维数，表示通过静态检测得到的抽象特征；$R = \{r_1, r_2, \ldots, r_m\}$，$m$ 为动态特征向量的维数，表示通过动态检测得到的抽象特征；$D=\{d_1, d_2, \ldots, d_t\}$，$t$ 为数据特征向量的维数，表示通过数据检测得到的抽象特征。

综合以上三个维度，隐私泄露事件定义可以扩充为：

$$E = \{(s_1, s_2, \cdots, s_n), (r_1, r_2, \cdots, r_m), (d_1, d_2, \cdots, d_t)$$

（2）

检测到的原始泄露事件由各种特征向量组成，需要通过评估模型进行隐私泄露评估，本文研究的具体泄露对象包括通讯录、短信、照片、定位信息等。不同的用户个体对各种隐私信息的关注程度会有所区别，例如有的用户认为通讯录信息比较重要，对地理位置的信息关注相对较少，而有的用户则会更加关注自己的地理位置信息。因此，隐私信息泄露是带有主观性的，不同用户根据自己对各种信息类型关注度的不同对隐私信息泄露行为做出不同的评价，在进行隐私泄露评估的时候需要考虑用户的主观因素。对本文研究的移动终端隐私泄露定义为：

定义 2：不符合用户主观意愿的读取移动终端隐私信息的行为即为隐私泄露。

基于此，将用户关注的敏感对象定义为 O：

$$O = \{o_1, o_2, \cdots, o_n\} \tag{3}$$

用户对敏感信息的重要程度有自己的预期，本文定义了五种级别 $L = \{0, 1, 2, 3, 4\}$，依次表示不关注、一般关注、关注、比较关注、特别关注。根据主观预期，用户判断相关敏感信息的泄露对自己会造成多大的影响，从而为其进行等级标记，构建了一个用户和敏感对象之间的关系 RL：

$$RL : O \rightarrow L \tag{4}$$

根据上述分析，对隐私泄露进行评估包含两个方面：一是检测读取用户隐私信息的行为；二是根据用户主观关注度进行量化评估。下文将分别对泄露事件的具体行为特征抽取和评估算法进行介绍。

4 隐私泄露行为特征抽取

Android 和 iOS 作为移动终端主流的操作系统，在运行环境和系统架构上有差异性，具体应用程序的行为特征也有区别，因此需要针对这两款操作系统分别进行泄露行为特征抽取的研究。

4.1 Android系统隐私泄露行为特征抽取

在 Android 系统架构下，隐私泄露行为的特征抽取包含三个部分：静态分析、动态分析和数据分析，如图 2 所示。其中静态分析和数据分析需要将应用程序包离线后在检测系统中进行分析，动态分析在修改了系统的 Android 终端上运行应用程序并进行相关检测操作。

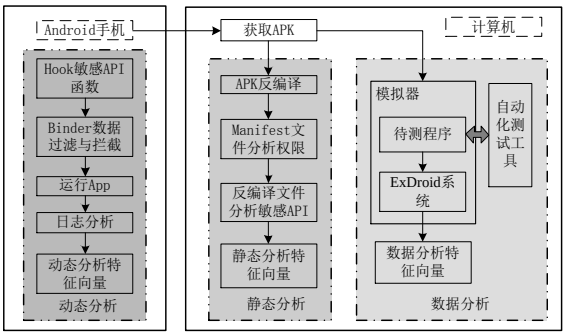

图 2 Android 分析系统实现架构

静态分析包括对 AndroidManifest 的分析和敏感 API 的分析两部分。其中 AndroidManifest 文件中的<users-permission>标签声明了应用安装时需要得到用户许可的权限，根据权限的具体功能和用户隐私泄露定义的相关性，本文选取了共 40 个权限，其中包含 19 个密切相关的权限和 21 个相关程度较低的权限。敏感 API 是应用程序调用隐私数据的接口，同样根据和用户隐私泄露定义的相关性，本文选取了 19 个隐私 API 函数以检测其调用情况。

综合 Android 系统的 AndroidManifest 文件分析和敏感 API 分析，静态分析特征向量 S_{Android} 表示如下：

$$S_{Android} = \begin{cases} s_{1-40} = SensitivePermission \\ s_{41-59} = PrivacyAPI \end{cases} \tag{5}$$

式中 s_i 的取值分别表示 AndroidManifest 文件的敏感权限分析（SensitivePermission）和 smali 代码的隐私 API 分析（PrivateAPI）结果，维数 n=59。

动态分析部分对应用程序的隐私泄露行为进行实时检测，检测内容共包含 8 类访问隐私数据行为，相关的隐私数据类型为：通讯录、IMEI、IMSI、通话记录、GPS 信息、本机号码、彩信、短信。动态分析的具体实现如图 3 所示，分为 Native C 层和 Java 应用层两部分。

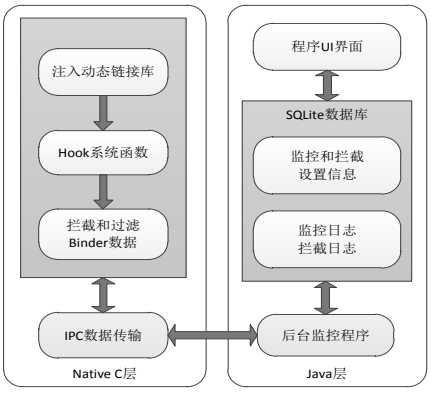

图 3 Android 动态分析架构

Native C 层实现了对系统关键函数的 Hook 以监控与隐私 API 接口相关的系统服务进程，包括 system_server 和 com.android.phone 进程，其中 system_server 的功能是提供位置、网络连接和 Android 框架系统服务等，com.android.phone 主要提供电信相关的服务，如通话、短信等[30]。Java 应用层主要负责监听 Native C 层上传的隐私泄露事件数据并进行存储，与用户的交互也在本层完成。

在 Android 动态分析中对 8 种泄露行为（访问通讯录、IMEI、IMSI、通话记录、GPS 信息、本机号码、彩信、短信）进行标记得到动态分析特征向量 R_{Android}：

$$R_{Android} = \{r \mid r = Action_i, i = 1,2,\cdots,8\} \tag{6}$$

该向量种的元素 r_i 的取值为 8 种 Action 分析结果，维数 m=8。

数据分析检测的实现主要基于污点跟踪技术，将应用程序安装到 ExDroid（Extended TaintDroid）系统中，对隐私泄露事件进行监控。为了克服 TaintDrdroid 标记不全、忽略 Native 代码的缺点[12]，本文对 TanintDroid 进行了扩展和改进实现了 ExDroid 系统，ExDroid 的简化模型如图 4 所示。

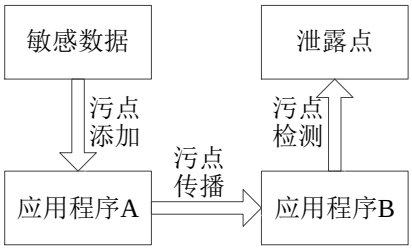

图 4 ExDroid 简化模型

首先对对敏感数据添加污点标签，该标签会随着敏感数据在系统中的流转而传播，最终在系统的出口点对污点标签进行提取来判断是否发生敏感数据泄露事件以及判断泄露行为类型。ExDroid 对 TaintDroid 定义的 17 种污点标签进行了扩展，将其增强到 23 种，扩展了对邮件、通话记录、多媒体文件、日历、系统设置和已安装应用的支持；在污点传播部分增加了对 Native 代码的获取支持，使得第三方动态库可以在系统中运行；在污点检测部分对网络、短信和文件三种泄露途径进行了分析。

数据分析共设置了 23 种污点数据标签，其特征向量 D_{Android} 的取值如下：

$$D_{Android} = \{N \mid N = Data(j)_p, \quad j = net, sms, file, p = 1,2,\cdots,23\} \tag{7}$$

其中 $Data(j)$，j=*net, sms, file* 表示污点数据泄露途径，$Data(j)_p$，j=*net, sms, file*, p=1, 2, …, 23 表示可能泄露的 23 种隐私数据类型，泄露途径有网络、短信和文件 3 种，检测结果共有 69 种可能性，因此元素 Nt 的维数 t=69。

4.2 iOS系统隐私泄露行为特征抽取

在 iOS 系统架构下，隐私泄露行为特征的抽取也分包含三个部分，实现框架如图 5 所示，其中静态分析包括对应用程序进行解密和反汇编，对反汇编后的代码进行敏感 API 遍历。动态分析需要挂钩敏感 API 函数以监控应用运行时的调用序列。数据分析在网络端口完成，对应用程序运行过程中产生的网络数据包进行抓取和分析。

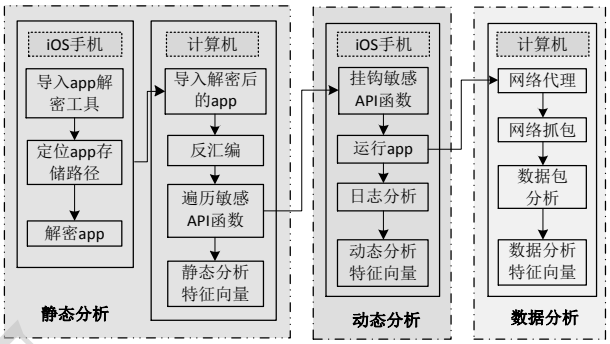

图 5 iOS 分析系统实现框架

通过对反汇编文件的敏感 API 调用序列的分析实现了 iOS 应用程序的静态特征向量和动态特征向量抽取[31]。根据 iOS 官方框架中对隐私数据类型的划分[32]，本文共定义了 9 种类型，每种包含了访问这类隐私信息的相关 API 函数，检测的 API 函数共有 20 个，如表 1 所示。

表 1 iOS 隐私 API 分类

隐私 API 分类	关键 API 函数举例	说明
定位服务 API（LocationApi）	requestWhenInUseAuthorization	程序前台运行时使用定位服务
通讯录 API（ContactsApi）	requestAccessForEntityType:(CNEntityType)entityTypecompletionHandler:((BOOL granted, NSError*error)) completionHandler	访问通讯录
日历和提醒事项 API（CalendarApi）	requestAccessToEntityType:(EKEntityType)entityTypecompletion:(EKEventStoreRequestAccessCompletionHandler)completion	访问日历和提醒事项

照片 API（PhotoApi）	requestAuthorization:(PHAuthorizationStatusstatus))handler	访问照片库的
麦克风 API（MicrophoneApi）	requestRecordPermission:(PermissionBlock)response	使用麦克风
相机 API（CameraApi）	deviceInputWithDevice:(AVCaptureDevice *)device error:(NSError **)outError	捕获设备输入
健康 API（HealthApi）	requestAuthorizationToShareTypes:(NSSet<HKSampleType *> *)typesToShare readTypes:(NSSet<HKObjectType *> *)typesToRead completion:((BOOL success, NSError*error))completion	访问健康数据
社交账户 API（SocialApi）	requestAccessToAccountsWithType:(ACAccountType *)accountType options:(NDictionary *)options completion:(ACAccountStoreRequestAccessCompletionHandler)completion	访问社交账户数据
本机信息 API（NativeinfoAPi）	extern NSString* CTSettingCopyMyPhoneNumber()	获取本机号码

由于 iOS 系统的应用程序文件都是经过 FairPlay[1] 加密的，在静态分析时首先需要对文件进行解密，解密后再调用 Hopper Disassembler 软件提供的相关函数对二进制文件进行反汇编，对汇编代码中的敏感 API 函数进行具体分析。iOS 静态分析模块检测了 20 个 API。因此，静态分析特征向量 S_{ios} 的取值为：

$$S_{iOS} = \{S \mid S = iOSPrivacyAPI_i, i = 1,2,\cdots,20\} \quad (8)$$

iOS 动态分析通过 MobileSubstrate[2] 框架编写 Tweak 插件实现，通过对隐私相关的 API 函数进行 Hook 以监控应用程序对其的调用行为，监控的 API 函数为 20 个，因此，动态分析特征向量 R_{iOS} 的取值为：

$$R_{iOS} = \{R \mid R = SysAPI_i, i = 1,2\cdots,20\} \quad (9)$$

数据分析特征向量通过在网络端口对应用程序运行过程中产生的数据包进行捕获和解析来抽取，如图 6 所示。如果数据已经通过 SSL 加密，可以通过挂钩 SSL 读写函数对加密前的数据进行捕获。获取数据包后解析得到详细的通信协议、目标 IP 地址和数据内容，根据分析结果判断是否存在泄露用户隐私数据的行为。

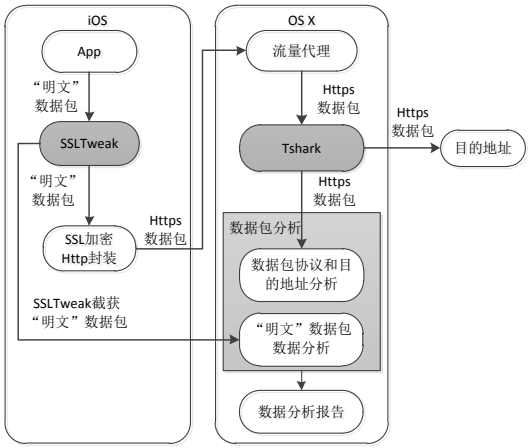

图 6 iOS 系统数据分析框架

结合网络端对网络协议和地址分析的结果，在数据分析方面共检测到四种泄露行为：

$Action_1$：应用将数据发送到非法地址，并且通过解析发送数据内容发现数据包含敏感信息。

$Action_2$：应用将数据发送到非法地址，但数据内容无法解析。

$Action_3$：应用未经用户授权将数据发送到应用的官方服务器，并且通过解析发送数据内容发现数据包含敏感信息。

$Action_4$：应用未经用户同意将数据发送到应用的官方服务器，但数据内容无法解析。

基于此，数据分析部分有 4 种泄露用户隐私数据的行为（$Action_i$，i=1,2,3,4），数据分析特征向量 D_{iOS} 的取值为：

$$D_{iOS} = \{D \mid D = Action_i, i = 1,2,3,4\} \quad (10)$$

5 基于用户主观预期的量化评估

在获取到隐私泄露事件数据后，需要结合用户的主观预期进行量化分析。前文已经针对用户对敏感对象的关注度进行了定义，描述了用户关注的敏感对象集合 O 和敏感级别 L，并构建了敏感对象和关注级别之间的关系 RL。接下来描述如何将检测到的事件进行解析从而得到与之对应的用户主观泄露级别。首先对事件的解析问题进行抽象：

1 FairPlay .https://en.wikipedia.org/wiki/FairPlay.2015.06.18

2 FREEMAN J. Mobile substrate. http://www.cydiasubstrate.com. 2016.02.14

- $E = \{e_1, e_2, \ldots, e_n\}$ 为检测到的隐私泄露事件集合，每个事件是一个向量。
- $T = \{s, r, d\}$ 表示检测分析的维度，分别为静态分析、动态分析和数据分析。
- $I = \{i_1, i_2, i_3\}$ 为各个维度在检测时占有的比重，反映了检测中对各个维度的侧重关系，相加等于 1。检测的维度包括静态检测、动态检测、网络数据检测。
- $EO = \{(eo_1, et_1), (eo_2, et_2), \ldots, (eo_n, et_n)\}$表示在 et_k 维度检测到敏感对象 eo_k 的泄露，其中 $eo_k \in O$，$et_k \in T$。
- $GetO: E \times O \rightarrow BOOL$ 是一个检测函数，如果 $o_j \in O$ 包含在检测事件 $e_i \in E$ 中则返回 True，否则返回 False。
- $GetT: E \rightarrow T$，获取检测事件所在的维度。
- $GetI: T \rightarrow I$，获取用户要求的检测比重函数。

对检测到的事件进行解析的算法如下：

算法. 检测事件解析算法.

输入：E, O

输出：EO

$EO \leftarrow \varnothing$

FOR $e \in E$ DO

  FOR $o \in O$ DO

    IF $GetO(e, o) = \text{True} \wedge (o, GetT(e)) \notin EO$

      $EO \leftarrow EO \cup \{(o, GetT(e))\}$

  END

END

RETURN EO

该算法的基本流程为：对检测事件进行遍历，如果发现其中有用户关注的隐私对象就将其加入到泄露对象集合中，并标注检测维度。

在对检测到的泄露事件进行解析得到集合 $EO = \{(eo_1, et_1), (eo_2, et_2), \ldots, (eo_n, et_n)\}$的基础上，综合考虑用户主观预期的影响，隐私泄露评估值为：

$$Risk = \frac{\sum_{i=1}^{m} RL(eo_i) GetI(et_i)}{\sum_{k=1}^{n} RL(o_n)} \times 100\% \quad (11)$$

其中，分子项为泄露对象的用户关注度和检测维度的比重相乘并累加，分母项为用户所有关注的敏感对象关注度的累加，除以分母项的用户对象关注度总和起到了归一化的效果。隐私泄露评估值越大，说明该应用涉及到的隐私泄露项目在用户定义的全部隐私泄露项目中的占比越高，对用户造成的潜在隐私泄露风险就越高。

6 测试结果与分析

6.1 隐私泄露行为检测

在原型系统中分别对 Android 和 iOS 应用进行了隐私泄露检测。在面对 SSL 加密数据时，由于 Android 检测系统采用是污点跟踪方法，定义的隐私数据没有被应用读取前已经被打上标签，在应用对数据进行处理后仍然可以在网络接口通过污点检测发现相关的敏感标记。而针对 iOS 平台，对于 SSL 加密前已经过应用程序私有加密的数据，无法解析出数据明文，但是可以解析出数据包使用的协议和数据包发送地址、目的地址。对于 SSL 加密前未经过应用程序私有加密的数据，iOS 应用在进行 https 加密传输时，使用的读写函数是 SSLRead 和 SSLWrite，因此可以通过对这两个函数挂钩来截获加密前的明文数据包内容，然后再通过捕获发送的数据包，解析出数据包使用的协议和数据包发送地址、目的地址，并和加密前明文数据进行匹配发现隐私数据泄露行为。

6.1.1 性能测试

分别针对 Android 平台和 iOS 平台进行相关功能和性能的测试。

Android 平台

通过五个应用程序的测试来验证系统功能的实现及执行性能对比，五个应用程序分别为 SdroidDemo、WeChat、AdobeReader、BaiduMap、taobao。其中 SdroidDemo 是专门编写的针对 Android 的测试程序，如图 7 所示。程序共包含 11 个 Activity 组件，MainActivity 下有十个 Button 控件，点击每个控件都可以实现一项敏感数据的传输，例如点击联系人，该应用会把联系人信息发送到指定位置。点击控件后进入的 Activity 均没有新的控件。

图 7 测试用例 SdroidDemo

首先进行静态分析的验证，对程序进行反编译获取敏感 API 调用信息，本系统的静态分析部分的结果与 Android 的 Kirin 框架的检测结果一致，除了检测 API 调用，本系统还在静态检测过程中分析了应用程序的 Activity 组件和敏感 API 关联情况，表 2 为静态分析的结果。

表 2 静态分析结果

应用	敏感 API 分析	Activity 总数	包含敏感 API 的 Activity 数
SdroidDemo	7 项 12 次	11	10
WeChat	21 项 123 次	198	69
AdobeReader	3 项 4 次	15	3
BaiduMap	12 项 37 次	59	21
taobao	25 项 77 次	287	92

将五个应用程序分别在本系统和目前广泛使用的 TaintDroid4.3 中运行，并开发了自动化检测模块通过计算机发送指令模拟人机交互，每个应用各执行三次取平均值。检测结果如图 8 和图 9 所示。

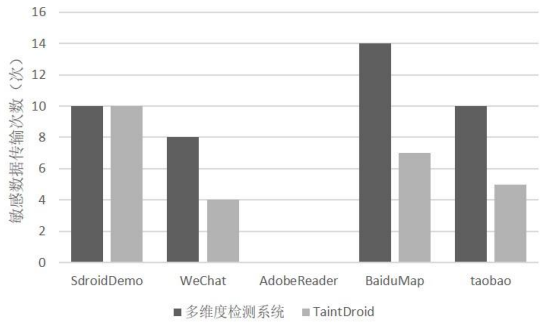

图 8 敏感泄露检出次数对比

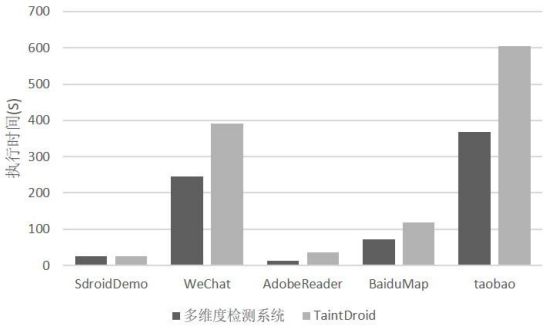

图 9 执行时间对比

从对比图中可以看出，本文的多维度检测系统能够对应用行为进行更加全面的分析，具有较高的检出率。在考虑测试效率的情况下，本系统在相对短的执行时间检测出应用程序的敏感传输，具有较好的检测效率。具体到单个应用，对于 AdobeReader，虽然执行时间大幅缩短，但由于并未检测到任何敏感传输，故不能说明通过静态分析执行路径指导动态执行的作用。对于 SdroidDemo 测试程序，由于本身功能和结构简单，检测过程中可以轻易实现完全覆盖所有可能，因此动静态结合执行作用不大。对于 BaiduMap，由于结构并不复杂，根据 Activity 调用图基本可以完全覆盖可能产生敏感传输的 Activity，动静态结合测试执行缩短了约 30%的执行时间。对于 WeChat 和 taobao，动静态结合测试执行缩短了约 40%的执行时间。

iOS 平台

针对 iOS 平台，考虑到平台的封闭性，在 AppStore 上下载四款已有应用进行测试，分别为 WeChat、AdobeReader、BaiduMap 和 taobao。静态检测得到的隐私泄露项目与 iOS 系统设置中的“隐私”选项中提示的内容完全一致。在界面组件和对应的功能上，应用的功能在 iOS 平台和 Android 平台并无差异，因此在两个平台上分析的界面组件总数和敏感 API 的对应关系保持一致。

将四款应用分别使用多维度检测系统和基于 MobileSubstrate 框架的 DiOS 进行动态执行过程的测试，按照用户使用习惯对应用进行交互执行，每个应用各执行三次取平均值。检测结果如图 10 和图 11 所示。

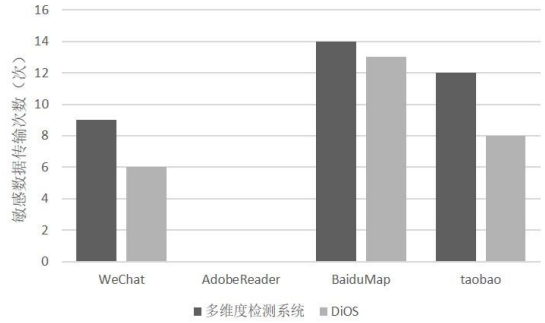

图 10 敏感传输检出次数对比

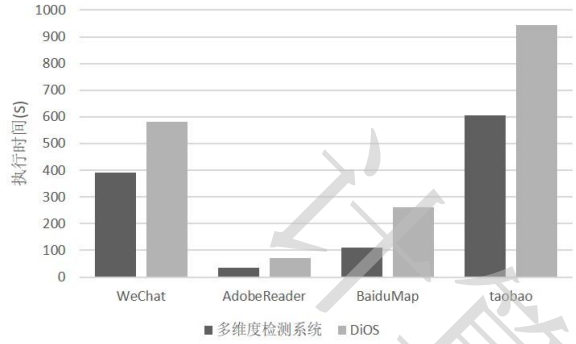

图 11 执行时间对比

在检测效果上，在 iOS 平台采用多维度检测系统同样检出率有所提升，并且检测所需的时间有所下降。

6.1.2 样本测试

本文从选取了在 Android 主流应用市场中各类别排名靠前的应用进行样本测试，包括 10 个类别，每个类别选取排名前 3 的应用，共计 30 个不同的应用程序。应用程序样本对静态、动态和数据分析向量中的 90%以上的行为特征进行了覆盖，其中大部分用户重点关注的隐私数据类型达到了完全覆盖，图 12 描述了根据 30 个应用的隐私泄露分析结果对目前普遍关注的 9 种隐私数据类型的汇总数据。

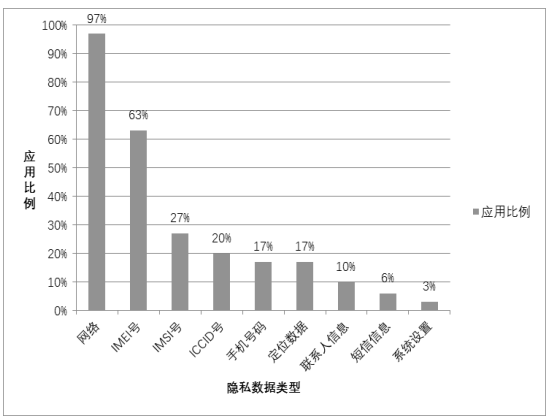

图 12 30 款 Android 应用程序分析结果

图 12 表明应用程序访问网络的行为占比最高，达到了 97%，只有一款单机运行的游戏应用没有申请访问网络的权限。IMEI 号作为终端的标识信息，泄露行为也比较普遍，约占 63%。有 5 款应用泄露了用户定位数据，但其中 4 款得到了用户的授权并将定位数据用于了定位相关的服务，1 款应用未得到用户授权就访问并发送了位置信息。手机号码、联系人、短信和系统设置都是比较私密的用户数据，也有不同程度的泄露，并且其泄露行为对用户具有较大影响。

本文从苹果的 AppStore 中同样选取了 10 个类别的排名前 3 的应用程序进行测试，共计 30 款不同的应用程序，并且覆盖了除“将数据发送至应用官方服务器，数据内容可以解析并包含敏感信息”外的所有行为特征，例外特征的出现是由于目前主流应用在将数据发送至官方服务器之前，都会对数据包进行重组、加密等处理，导致无法从网络数据包中解析信息。应用行为特征检测结果如图 13 所示。

图 13 中涵盖了 9 类敏感 API 的分析结果，API 检测结果汇总了静态和动态分析的数据，网络数据分析结果 N1、N2、N3、N4 汇总了在网络端口进行分析得到的数据。由图 13 可以看出泄露用户位置信息占比最高，达到了 73.3%，访问通讯录、相册和相机的应用占比也较高，超过了 30%以上，通过网络端的数据分析发现 46.7%的应用程序都未经用户授权将数据发送到应用服务器，但由于进行了数据加密，难以解析出发送内容。

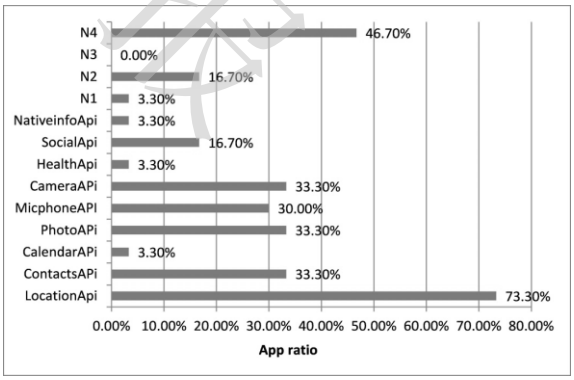

图 13 30 款 iOS 应用程序分析结果

综合检测结果进行分析，约有 60%以上的应用在网络端发送数据时候采取了 SSL 加密。而应用程序采用私有方式对数据进行重组包、加密等改变原有数据格式时，应用对不同的数据会做不同的处理，大部分应用程序在涉及到用户消息传输的时候基本都做了加密处理，例如社交类软件传输用户发

送的信息；约 75%的应用在发送其他信息时没有在程序体内处理，例如 IMEI、IMSI 微博搜索的关键字等。在数据分析过程中遇到加密信息时，如果应用采用的是系统提供的 API（例如加密、hash），可以与动态分析相结合对数据进行匹配；应用如果没有采用系统提供的相关 API 对原始数据做处理，数据则难以被解析和匹配。

6.2 隐私泄露评估值计算

在获取隐私泄露事件后，对隐私泄露进行量化评估需要设定两个值，一是各种维度的检测方式占有的比重；二是用户对泄露对象的关注度。

在对三个维度检测比重的设置方面，考虑到静态分析是事前检测，代码中扫描到访问敏感 API 的行为并不等同于实际调用的发生，因此静态分析占最终判断隐私泄露的权重较低，本文设定为 0.1；动态分析检测到的行为表示实际发生的泄露事件，因此其权重值高于静态分析，本文设定为 0.3；数据分析由于在移动终端边界进行检测，检测到的事件表示应用已经读取了隐私数据并发送出去，造成隐私泄露影响力较大，因此本文设定为 0.6。

在用户对泄露对象的关注度方面，本文列举了 9 种敏感客体：位置、网络、电话、联系人、短信、设备、麦克风、相机、照片，敏感对象集合为 $O = \{o_1, o_2, \cdots, o_9\}$, o_i, $i = 1, 2, \cdots, 9$ 分别与上述敏感客体对应。在实际应用中，可以通过用户填表、搜集用户初始设置等方式进行对敏感信息关注度的提取。本文假设有用户 A 和用户 B，两个用户对敏感信息的关注度设置如图 14 所示，其中用户 A 主要关注点在短信和联系人，用户 B 主要关注点在相机和麦克风。

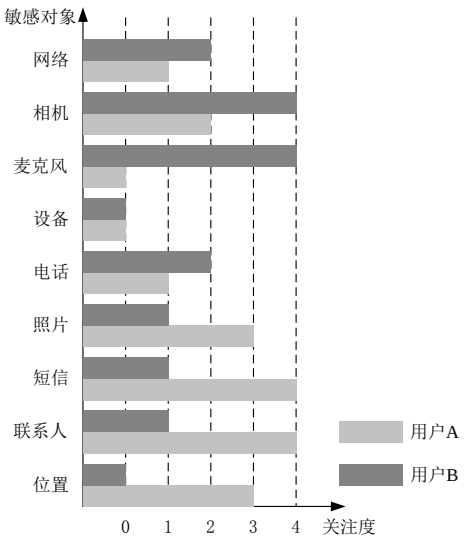

图 14 用户关注度设置

通过对一款 Android 新闻类应用程序 X 进行分析，详细阐述隐私泄露评估值的分析和计算过程，该应用程序的隐私泄露事件分析如表 3 所示。

表 3 应用程序 X 的分析结果

分析方法	行为特征	特征备注	
静态分析	SensitivePermission	ACCESS_NETWORK_STATE	访问网络信息
		INTERNET	打开网络套接字
		ACCESS_WIFI_STATE	访问 WiFi 状态
		CHANGE_NETWORK_STATE	改变网络连接状态
	SensitivePermission	READ_PHONE_STATE	读取手机运行状态
	PrivateAPI	ContentResolver;->query	读取联系人、短信数
		HttpClient:->execute	发送 HTTP 请求
		TelephonyManager;->getCellLocation	获取手机所在小区信息
		TelephonyManager;->getDeviceId	搜集手机 IMEI 码
		TelephonyManager;->getSubscriberId	获取手机 IMSI 码
动态分析	Action1	IPhoneSubInfo	获取 IMEI 号
	Action2	IPhoneSubInfo	获取 IMSI 号
	Action4	IActivityManager	访问短信
	Action5	IActivityManager	访问通讯录

	Action6	IActivityManager	访问彩信
	Action8	ILocationManager	访问 GPS 信息
数据分析	File	LEAKAGE_IMEI	泄露 IMEI
		LEAKAGE_CONTACTS	泄露联系人
		LEAKAGE_SMS	泄露短信
		LEAKAGE_LOCATION_NET	泄露位置
		LEAKAGE_IMSI	泄露 IMSI
	Net	LEAKAGE_IMEI	泄露 IMEI
		LEAKAGE_IMSI	泄露 IMSI

对检测事件进行解析后，得到 $EO = \{(o_2, s), (o_6, s), (o_5, s), (o_4, s), (o_1, s), (o_6, r), (o_5, r), (o_4, r), (o_1, r), (o_6, d), (o_4, d), (o_5, d), (o_1, d)\}$，最后得到用户 A 对应用程序 X 的隐私泄露风险评估值为：

$$Risk = \frac{\sum_{i=1}^{m} RL(eo_i)GetI(et_i)}{\sum_{k=1}^{n} RL(o_n)} \times 100\%$$

$$= \frac{12.7}{18} \times 100\% = 70.56\%$$

用户 B 的泄露风险评估值为：

$$Risk = \frac{\sum_{i=1}^{m} RL(eo_i)GetI(et_i)}{\sum_{k=1}^{n} RL(o_n)} \times 100\%$$

$$= \frac{3.3}{15} \times 100\% = 22\%$$

可以看出由于这款新闻类应用程序对短信和联系人信息在不同维度进行了泄露，因此用户 A 对其泄露风险的评估较高。而用户 B 关注点在于麦克风和相机的使用情况，这款应用并未发生此类泄露，因此对其泄露风险的评估值较低。

对 iOS 的一款社交类软件 Y 进行评估，其具体检测结果如表 4 所示：

表 4 应用程序 Y 的分析结果

分析方法	行为特征	
静态分析	LocationApi	定位服务
	ContactsApi	通讯录
	PhotoApi	照片
	MicrophoneApi	麦克风
	CameraApi	相机
动态分析	LocationApi	定位服务
	ContactsApi	通讯录
	PhotoApi	照片
	MicrophoneApi	麦克风
	CameraApi	相机
	NativeinfoApi	本机信息
数据分析	行为（4）	加密发送信息到服务器

在数据分析种，由于采用了加密手段导致无法解析恢复泄露数据明文，因此假设其在动态分析中读取的内容都存在数据分析种泄露的风险。针对用户 A 计算的泄露风险评估值分别为：12/18*100%=66.67% ，用户 B 为：10/15*100%=66.67%.由于这款游戏应用对短信、通讯录、相机和麦克风都在各个维度进行了访问，用户 A 和 B 对其的评估值恰好相近。

由上述分析可以看出，基于用户主观预期的评估模型结合多个检测维度对应用程序的隐私泄露行为进行评估，能够反映用户对隐私对象的关注度设置。

7 总结

本文针对移动终端隐私泄露检测问题，提出了多维度的检测框架，综合静态分析、动态分析和数据分析三种方法，为 Android 和 iOS 终端提供全面的检测方案。在多维度框架下对检测事件进行抽象，分别为两种终端平台提供了隐私泄露特征抽取方法。在隐私泄露评估种引入了用户主观预期的概念，结合用户自身特点进行量化。实验结果表明，本文提出的框架能够有效的检测应用隐私调用情况，评估模型能够有效体现用户自身对隐私对象评价的主观性。

本文目前的主观评价工作是假设已经获取用户关注度的基础上开展的，在后续的研究工作中，需要进一步研究用户关注度的抽取。另外在隐私泄露事件的检测中，目前静态分析和动态分析方法相对完善，数据分析部分的准确度还有待进一步提高，尤其是面对加密数据的时候，需要和静态、动

态分析结合，针对应用对数据的操作进行分析，使用更加有效的方法获取传输内容。

参 考 文 献

[1] Peng Li, Baojiang Cui. A comparative study on software vulnerability static analysis techniques and tools// Proceedings of the 2010 IEEE International Conference on Information Theory and Information Security. Beijing, China, 2010: 521-524

[2] Yang Guang-liang, Gong Xiao-rui, Yao Gang, Han Xin-hui. A privacy leakage detection system for Android. Computer Engineering, 2012, 38(23): 1-6
(杨广亮, 龚晓锐, 姚刚等. 一个面向Android的隐私泄露检测系统. 计算机工程, 2012, 38(23): 1-6)

[3] M. Apel, C. Bockermann, M. Meier. Measuring similarity of malware behavior // Proceedings of the IEEE 34th Conference on Local Computer Networks. Piscataway, Zurich, Switzerland, 2009: 891-898

[4] W. Enck, M. Ongtang. P. McDaniel. On lightweight mobile phone application certification // Proceedings of the 16th ACM Conference on Computer and Communications Security(CCS'09). Chicago USA, 2009: 235-245

[5] Tong Zhen-Fei. A static behavior-based method to detect malware on Android [Mater's Thesis]. Nanjing University of Posts and Telecommunications, Nanjing, 2012
(童振飞. Android恶意软件静态检测方案的研究[硕士学位论文]. 南京邮电大学计算机学院, 南京, 2012)

[6] M. Nauman. S. Khan, X. Zhang. Apex: extending Android permission model and enforcement with user-defined runtime constraints// Proceedings of the 5th ACM Symposium on Information,Computer and Communications Security. Beijing, China, 2010: 328-332

[7] S. Arzt, S. Rasthofer, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps // Proceedings of the 35th annual ACM SIGPLAM Conference on Programming Language Design and Implementation, Edinburgh, UK, 2014: 1-11

[8] A. P. Fuchs, A. Chaudhuri, and J. S. Foster. Scandroid: Automated security certification of android applications. Maryland, USA: University of Maryland, Technical Report: CS-TR-4991, 2009

[9] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang. Appintent: Analyzing sensitive data transmission in android for privacy leakage detection // Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, Berlin, Germany, 2013: 15-27

[10] Manuel Egele, Christopher Kruegel, Engin Kirda. PiOS: detecting privacy leaks in iOS applications// Proceedings of the Network and Distributed System Security Symposium. California USA, 2011: 15-30

[11] A. M. Daniel. Idb: A tool for blackbox iOS security assessments // Proceedings of the 2016 IEEE International Conference on Mobile Software Engineering and Systems, Austin, USA, 2016: 181-282

[12] W. Enck, P. Gilbert, B. Chun. TaintDroid: an information flow tracking system for real-time privacy monitoring on smartphones. Communications of the ACM, 2014, 57(3): 99-106

[13] Peter Hornyack. These aren't the droids you're looking for: retrofitting Android to protect data from imperious application// Proceedings of the 18th ACM Conference on Computer and Communications Security(CCS'11). Chicago, USA, 2011: 213-221

[14] Zhenyu Ni, Ming Yang, Zhen Ling, Jia-nan Wu, and Junzhou Luo. Real-time detection of malicious behavior in Android apps // Proceedings of 2016 International Conference on Advanced Cloud and Big Data, SiChuan, China, 2016: 221-227

[15] Yeh Chao-chun, Lu Han-Lin, Chen Chun-yen. CRAXDroid: Automatic android system testing by selective symbolic execution // Proceedings of 2014 Eighth International Conference on Software Security and Reliability, Redwood City, USA, 2014: 140-148

[16] Peter Gilbert, Byung-Gon Chun, P. Landon. Automating privacy testing of smartphone applications. North Carolina: Duke University, Technical Report: CS-2011-02, 2011

[17] Y. Agarwal, M. Hall. ProtecMyPrivacy: Detecting and mitigating privacy leaks on iOS devices using crowdsourcing // Proceedings of the 11th annual International Conference on Security and Cryptography, Taipei, China, 2013: 110-125

[18] Andreas Kurtz, Andreas Weinlein, Christoph Settgast. DiOS: dynamic privacy analysis of iOS applications. Freistaat Bayern: Friedrich-Alexander-Universität Erlangen-Nurnberg, Technical Reports: CS-2014-03, 2014

[19] J. P. Achara, A. Francillon. MobileAppScrutinator: A simple yet efficient dynamic analysis approach for detecting privacy leaks across mobile Oss. New York,USA: Cornell University, Technical Reports: arXiv:1605.08357, 2016

[20] M. Pistoia, O. Tripp, P. Centonze, J. W. Ligman. Labyrinth: Visually configurable data-leakage detection in mobile applications // Proceedings of the 16th IEEE International Conference on Mobile Data Management, Washington, USA, 2015: 279-286

[21] T. Omer, R. Julia. A Bayesian approach to privacy enforcement in smartphones // Proceedings of the 23rd USENIX Conference on Security Symposium, San Diego, USA, 2014: 175-190

[22] X. Y. Zhou, S. Demetrious, D. J. He, et al. Identity, location, disease and more: Inferring your secrets from android public resources // Proceedings of the ACM Conference on Computer and Communications Security, Berlin, Germany, 2013: 1017-1028

[23] A. Nadkarni, W. Enck. Preventing accidental data disclosure in modern operating systems // Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, New York, USA, 2013: 1029-1042

[24] S. Chen, R. Wang, X. F. Wang, K. Zhang. Side-channel leaks in Web applications: a reality today, a challenge tomorrow // Proceedings of the 2010 IEEE Symposium on Security and Privacy, Oakland, USA, 2010: 100-116

[25] M. Theoharidou, A. Mylonas, D. Grizalis. A risk assessment method for smartphones // Proceedings of the IFIP International Information Security Conference, Ghent, Belgium, 2012: 443-456

[26] W. Yan, W. Guolin. An evaluation model for information security of android application based on analytic hierarchy process // Proceedings of the 2016 World Automation Congress, Guilin, China, 2016: 1-6

[27] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie. WHYPER: Towards automating risk assessment of mobile applications // Proceedings of the 22nd USENIX Security Symposium, Washington, USA, 2013: 571-584

[28] Y. Jing, G. Ahn, Z. Zhao, and H. Hu. RiskMon: Continuous and automated risk assessment of mobile applications // Proceedings of the 4th ACM Conference on Data and Application Security and Privacy, Texas, USA, 2014: 205-230

[29] Y. Jing, G. Ahn, Z. Zhao, and H. Hu. Towards automated risk assessment and mitigation of mobile applications. IEEE Transactions on Dependable and Secure Computing, 12(5), 2015: 571-584

[30] Zeng Yang. Design and implementation of Android privacy protecting software [M.S. Thesis]. Beijing University of Posts and Telecommunications, Beijing, 2013
(曾阳. Android手机隐私保护软件的设计与实现[硕士学位论文]. 北京邮电大学, 北京, 2013)

[31] M. Epifani, S. Pasquale. Learning iOS Forensics [[M.S. Thesis]. Birmingham: Packt Publishing Ltd, 2015: 153-170

[32] Yuvraj Agarwal, Malcolm Hall. Protect my privacy: detecting and mitigating privacy leaks on iOS devices using crowdsourcing// Proceedings of the 11th annual international conference on mobile systems, applications, and services. Taipei, China, 2013: 249-262

LI Tao, born in 1984, Ph.D., Assistant Professor.His Research interests include mobile terminal security, privacy protecting, trusted computing.

Wang Yongjian, born in 1981, Ph.D., , Associate researcher, His Research interests includes information security, data analysis etc.

Xing Yuexiu, born in 1991, PH.D. candidate. His Research interests include privacy protection, mobile terminal forensic.

Hu Aiqun, born in 1964, Ph.D., Professor. His Research interests include network and information security, physical layer security.

Background

This paper focuses on the privacy leakage testing for mobile terminals. Although there are many researches of this area in the worldwide, these researches are mainly in leakage event testing. What should we do after acquiring the leakage data? Efficient analysis method based on leakage event data is lacked. Even in the existing leakage testing researches, only one-dimension is concerned. Both the static analysis and dynamic analysis have their own limitation. More comprehensive data collection framework is also required.

This paper proposed an evaluation model that creatively introduced user's subjective anticipation to leakage quantification. The evaluation model with subjective feather provides a new method for privacy leakage analysis. As to the leakage event, we collected three-dimension data, which provided more complete original data for evaluation model.

Our topic belongs to the project of National Natural Science Foundation of China. The project name is "Intelligent Security Mechanism Research of Information System Based on

Quantitative Trusted Model (61601113)". This project proposes an intelligent security concept for information system based on basic theory of active defense and trusted computing. We aim to solve several key problems in the intelligent trusted mechanism: 1) Through the combination of related theory about information system structuration and trusted computing, we propose a hierarchical package constructing method for trusted model, which will provide supporting for the implementation of trusted mechanism. 2) For the problems of quantitative analysis in existing trusted evaluation, we propose a quantitative method to compute trusted value based on fuzzy theory. An adaptive adjustment process is also presented through the feedback of quantization value, which can be used to accomplish intelligent security mechanism. 3) Aims at the problems of non-interference information flow model, we propose a quantitative method for privacy leakage based on Shannon entropy theory, which will realize dynamic control for privacy protecting. 4) Aims at the requirements of qualitative analysis for integrity protecting, we propose an abstract method for access path based on graph theory. Through the probabilistic forecasting, we will quantitatively calculate integrity and provide basis for access control deciding. Through researching the above problems, we will improve theoretical framework of intelligent security and construct implementation method of intelligent security mechanism based on basic theory of trusted computing.

This paper captures the leakage data of applications and evaluates subjective leakage value. The value is the inputs of automatic adjustment mechanism. If the value is high, the behaviors of application will be restricted automatically.

Our research group has acquired impressive results on privacy leakage evaluation. Our project "Research and Application of Key Technologies of Security Testing for Mobile Terminals" won the second award by State Quality Inspection Administration. We also published 5 international journal papers in the area of trust evaluation for mobile terminals.