基于 Android 无障碍服务的行为监控

马文聪[1],谭毓安[2],冯 硕[1],刘 璐[1],李元章[1]
(1. 北京理工大学计算机学院,北京 100081;2. 北京理工大学网络空间安全学院,北京 100081)

摘 要: 用户在手机上的异常行为给社会、企业和个人带来一定的损失和风险. 例如用户使用手机违规记录企业的敏感信息、使用手机终端在社交网络上散布违法违规内容和言论等. 然而目前尚没有直接运行在终端、对用户本机应用操作进行监控的技术. 鉴于目前大部分手机都是 Android 平台,本文以 Android 智能手机为研究对象,提出一种基于无障碍服务的用户行为监控技术. 用户界面发生变化时,会产生大量基于视图层次结构编写的无障碍事件. 本方法筛选出关键的无障碍事件并对其进行遍历,获取界面组件元素、无障碍事件的类型、界面焦点对象等信息,进而判断用户是否存在敏感行为. 本方法不依赖于特定的 Android 版本;通过对无障碍事件进行过滤,提高处理性能;通过调整应用监控范围和监控粒度,保障用户隐私. 为了证明本方法的可行性,在真实 Android 设备上进行测试,可正确监控用户在四种不同应用上的行为. 性能测试表明本方法的平均延迟小、CPU(Central Processing Unit)占有率低、内存消耗少,不影响用户正常使用.

关键词: Android;无障碍服务;用户行为;监控;隐私;UI 界面
基金项目: 国家自然科学基金(No.62072037,No.U1936218)
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2023)12-3572-10
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220319

Behavior Monitoring Based on Android Accessibility Service

MA Wen-cong[1], TAN Yu-an[2], FENG Shuo[1], LIU Lu[1], LI Yuan-zhang[1]
(1. *School of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081, *China*;
2. *School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing* 100081, *China*)

Abstract: The abnormal behavior of users on mobile phones brings certain losses and risks to the country, society, enterprises and individuals. Such as, users record sensitive information of enterprises with the help of mobile phones, or use mobile phones to spread false statements on social networks. However, there is no technology to directly run on the terminal to monitor the user's local application operation. Since most mobile phones are Android platforms, this paper takes Android smart phones as the research object, and proposes a user behavior monitoring technology based on accessibility service. When the phone's interface changes, there will be a large number of accessibility events written based on the view hierarchy. Our method selects the key accessibility events, traverses them, obtains the information on interface component elements, types of accessibility events, interface focus objects and so on, and then judges the user's behavior. Our method does not depend on a specific Android version; improves processing performance by filtering accessibility events; protects users' privacy by adjusting the application monitoring scope and monitoring granularity. In order to prove the feasibility of our method, we experiment on the devices in the physical world and correctly monitor the behavior of users in four different applications. Performance testing shows that our method has low CPU consumption and low average memory occupancy.

Key words: Android; accessibility service; user behavior; monitor; privacy; UI interface
Foundation Item(s): National Natural Science Foundation of China (No.62072037, No.U1936218)

1 引言

Android 作为当前最为流行的移动智能操作系统,设备和用户数量庞大,应用程序丰富. 2021 年 Google Play 上的应用已经超过 260 万种[1]. 随着 Android 的飞速发展,出现了许多与 Android 有关的安全漏洞和隐私泄漏问题[2],如 Android 驱动漏洞问题[3]、第三方登录服务中身份信息泄露问题[4]、第三方 SDK(Software Devel-

收稿日期:2022-03-28;修回日期:2022-07-12;责任编辑:梅志强

opment Kit)漏洞[5]等. Android系统的安全问题越来越受到重视,各种各样的检测和监控方法被提出,例如在不执行应用程序的情况下,通过反汇编分析应用源代码来检查应用是否具有恶意功能,常见的技术手段有基于特征的检测[6,7]、基于权限的检测[8]、基于基于诱导机制的检测[9]、基于Dalvik字节码分析的检测[10]. 还可以通过监控应用的运行[11,12],如数据流动是否正常、设备状态是否正常等等,判别是否为潜在的恶意应用.

很多人意识到恶意应用[13]带来的隐私泄露问题,却忽视了用户本身操作带来的隐私泄漏. 比如,用户是否通过社交软件发布涉密信息. 用户在涉及隐私的场景下,违规使用拍照、录音等功能. 随着手机和网络的快速发展,用户因为有意或无意的违规操作带来的涉密信息泄漏问题越来越严重. 因此,对于用户行为本身的监控是非常具有意义的.

从手机制造厂商视角设计并开发的用户行为监控系统,是不具备可移植性的. 一个厂商的监控系统无法移植到另一个厂商的手机里面. 同时,我们也无法要求让应用本身监控用户在自身上的操作.

本文针对上述局限性,利用无障碍服务[14]设计了一种应用进行用户行为监控,它能够部署在各种不同手机厂商的设备上. 该监控程序遍历基于视图层次结构编写的无障碍事件,根据界面组件元素、无障碍事件的类型、界面焦点对象等信息,判断出用户的行为. 对于敏感的用户行为,将用户操作的具体细节存放到数据库中. 相关人员可以进一步通过分析这些记录,判断用户是否存在违规操作.

本文的主要创新点与贡献如下:

(1)以黑盒模式对用户行为进行监控,无需提前知道任何Android系统信息、应用信息、设备信息,也不会干扰应用在运行过程中的数据流通.

(2)首次提出利用无障碍服务对用户行为进行监控和分析,不需要获取root权限,也不需要对Android系统和被监控应用进行任何修改. 作为后台应用直接部署在手机上,操作简单.

(3)本文在Galaxy S10和Mi 10设备上,对四种常见应用上的用户行为进行了实际验证,实验结果表明本文提出的方法具有较好的兼容性.

2　相关工作

2.1　用户行为监控

现有的关于用户行为监控的思路有两种:一种是通过对UI(User Interface)界面进行监控,推断用户行为;另一种是通过hook技术,捕获其他应用的数据流,推断用户行为.

Android自身提供的UI Automator可以从Android设备请求视图层次结构,但返回数据需要延迟[15],不具备实时性. 文献[16]通过对应用的源码进行修改,插入一个用于捕获UI事件的库,让应用本身记录用户行为. 这需要耗费额外的时间和性能来修改大量应用的源码. 文献[17]对用户在网络应用程序上的行为进行监控. 通过代理服务器将JavaScript记录器注入到网页上,这样就能够记录用户在网页上的行为. 然而该方法依赖于代理服务器,只适用于Web应用程序,而不适用于本机移动应用程序. 文献[18]设计了新的能够实时动态捕获UI数据的系统. 它包括3个组成部分,即Android手机设备、后端服务器和Web浏览器. 安卓设备需要运行安卓操作系统的修改版本,该版本实现了一种定制服务:使用自定义方法扩充定义UI和UI元素的Android类,并以可扩展标记语言(eXtensible Markup Language,XML)格式返回UI的分层表示,通过USB(Universal Serial Bus)传送给后端服务器. 后端服务器再将信息以jpeg的格式发送给客户端Web界面. 用户需要通过Web界面和应用程序进行交互,这样系统才能捕获用户行为和界面信息. 类似通过Web界面进行交互的思路还有文献[19]. 很显然上述技术在实际使用中无法对用户行为进行监控. 我们提出的解决方案可以将监控系统作为普通应用程序进行安装,不需要借助第三方设备监控用户对手机的操作,也不需要对Android系统和被监控应用进行任何修改,部署简单便捷.

通过hook技术[20,21]可以直接在手机上进行用户行为监控. 在Android系统中,应用程序,包括应用触发事件和后台逻辑处理,都是根据事件流一步步地执行. 通过hook技术可以在函数层面对应用执行的数据流进行拦截,分析应用的行为、获取用户数据并进行处理. 然而上述技术的使用场景存在限制. 由于每个应用存在自身独立的进程空间,所以hook必须拥有root权限,才能注入到应用所在的进程空间. 随着Android漏洞不断减少,通过漏洞获取root权限的难度越来越大;除此之外,手机厂商对于Secure Boot的实现不断增强,很多机型不能通过官方渠道解锁bootloader,通过刷第三方rom或recovery来获取root权限的方式逐渐失效. 另外一些监控系统通过提供虚拟环境对应用进行hook[22],让应用运行在自己的沙箱中,自己本身充当Android系统和应用之间的媒介. 虚拟环境模拟Android系统,对系统服务实现代理和替换,这面临着Android版本兼容性问题. 随着Android版本的更迭,虚拟环境要随之迭代更新,极大地增加了工作量. 我们提出的解决方案不依赖于特定机型,不需要获取root权限,也不依赖于特定的Android版本,具有通用性和鲁棒性.

2.2 无障碍服务

Android系统在发展过程中,功能不断丰富完善,提高了用户体验度. 无障碍服务(accessibility service)在Android 1.6版本中被提出,在设计之初是为了帮助残障用户或可能暂时无法与设备进行全面互动的用户完成操作. 比如放大屏幕上的字体,将文字转化为语音及进行自动点击等[23]. 自Android 4.0开始,Android系统基于视图层次结构编写无障碍事件,极大地增加了无障碍服务可以获取的有关界面互动的信息量,打破了Android系统提供的沙箱限制[24,25].

无障碍服务的主要用途是协助残障人士更方便地使用手机,例如,屏幕阅读器应用程序会大声读取触摸按钮上的文本标签,以帮助有视觉障碍的用户使用该设备[26,27]. 但也有一些攻击者利用无障碍服务攻击用户系统,例如记录敏感用户输入、从其他应用窃取私人数据,或基于静默安装进行注入[28].

还有不少工作利用无障碍服务辅助完成其他操作,如借助无障碍服务对UI界面进行监控. 文献[29]利用无障碍服务访问设备上的其他程序,加速GUI(Graphical User Interface)测试. 文献[30]提出一种利用无障碍服务捕获UI交互,自动生成鲁棒的、性能良好的软件测试的方法. 文献[31]利用无障碍服务识别和操作设备界面,设计了一种自动控制工具来减少用户的重复操作. 文献[32]通过监控UI界面判断是否存在恶意应用通过构建可能导致信息泄漏的UI交互图来泄漏私人信息,这项工作仅仅是用于监控软件的恶意行为,而不是对用户行为本身进行监控. 文献[33]基于无障碍服务设计了一款基于主机的SDN(Software Defined Networking)代理工具,监控用户触发的网络数据流,但是这项工作仅仅通过无障碍服务返回的控件ID(Identity document)获取控件信息,对于那些没有ID的控件,无法获取相关信息. 除此之外,这项工作通过截取网络流对用户上网行为进行监控,无法对用户使用本机移动程序行为进行监控.

3 基于无障碍服务的用户行为监控需求分析

3.1 监控范围判定

不同用户,手机上的应用不尽相同. 我们以系统应用文件管理助手、相机、录音应用以及市场热门应用QQ、微博、Gmail为例,进行用户行为分析. 在系统自带的文件管理助手中,用户可以对文件进行增删改查以及文件位置的移动;在拍照应用中,用户可以进行拍照;在录音应用中,用户可以进行录音;热门应用QQ,是一款即时通信工具,用户可以发送文本信息和文件,同时还可以拍照和语音;Gmail,是一款电子邮件服务软件,用户可以发送邮件和附件;微博,是一款社交媒体应用,用户可以分享文字、上传图片和视频.

用户泄漏隐私、发布不实信息的媒介一般只有文字、图片、语音和文件. 本文提出的监控框架,对于文本信息,能记录用户输入的文字内容以及用户是否发送了文本信息;对于图像信息,能监控用户是否存在拍照和发送图片行为;对于语音信息,能监控用户是否录音、是否发送录音文件;对于文件,能监控用户对于文件的增删改查以及是否发送. 用户敏感行为的界定标准如表1所示.

表1 用户敏感行为界定标准

用户行为	监控	界定标准
发送消息	文字内容	设定敏感词,通过无障碍服务API(Application Programming Interface)获取文字内容,判断敏感词是否被包含在内
拍照	相机启动	通过谷歌API geolocation获取经纬度信息.在特定地理位置范围内的拍照活动视为敏感行为
录音	录音启动	通过谷歌API geolocation获取经纬度信息.在特定地理位置范围内的录音活动视为敏感行为
文件	增删改查	设定敏感词,通过无障碍服务API获取内容,判断敏感词是否包含在内

3.2 无障碍事件分析

如图1所示,无障碍服务充当普通应用程序和本方法之间的中介[34]. 当开启无障碍服务后,本方法会在后台静默运行,一旦本方法注册过的无障碍事件到来,系统会回调本方法的无障碍事件处理函数onAccessibilityEvent(),在这个函数中,系统将基于视图层次结构编写的无障碍事件作为参数传递进来,视图层次结构包含该组件的一系列界面组件(其父级)和该组件可能包含的界面元素(其子级). 本方法遍历无障碍事件,获取一系列界面组件元素、无障碍事件的类型、界面焦点对象,进而判断出用户的操作信息.

(1)界面组件元素:例如,如果无障碍事件中存在输入框控件,我们可以进一步判断出用户正在打字,并能获取用户输入的文字;如果无障碍事件中存在文本框控件,我们可以进一步判断出用户正在浏览文本信息,并能监控用户浏览的内容. 值得注意的是,在关注控件本身信息的同时,我们需要关注到控件的上下文信息.

(2)无障碍事件类型:例如,当收到的无障碍事件类型为TYPE_VIEW_CLICKED,说明用户在屏幕上进行点击;当收到的无障碍事件类型为TYPE_VIEW_TEXT_CHANGED,说明用户正在输入文字;当收到的无障碍事件类型为TYPE_VIEW_SCROLLED,说明用户

在滑动屏幕.

(3)界面焦点对象:界面焦点改变往往意味着用户从一个操作状态进入了另一个操作状态.

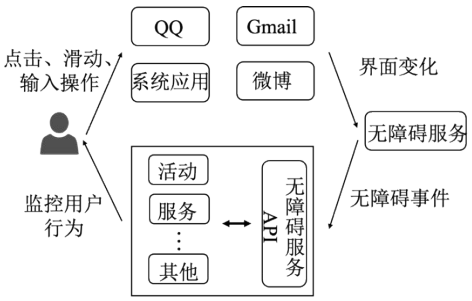


图1　Android无障碍服务框架

以QQ的文字聊天功能为例,用户滑动聊天窗口、在输入框中打字、点击发送按钮,都会对界面产生影响.无障碍服务以无障碍事件的形式将整个操作中产生的数据流传递给后台运行的监控服务.监控服务对无障碍事件进行分析提取,判断出用户的行为.

3.3　技术难点与解决思路

为了监控用户行为,本方法需要长期运行在后台,这对性能开销提出了挑战.如果频繁响应无障碍事件,不仅会影响用户的前台体验,而且会带来过多的资源开销.由于本方法需要对用户在本机上的所有操作进行监控,无法根据无障碍事件的包名对无障碍事件进行过滤.为了降低性能开销,首先,对监听的无障碍事件进行过滤,仅仅注册关注的无障碍事件.其次,无需对用户单一操作带来的多个无障碍事件进行重复处理,通过适当延迟来统一处理同时到来的无障碍事件.

需要减小系统版本更新、应用版本更新对本方法带来的影响.为了降低对系统版本的依赖,本方法仅仅使用系统提供的无障碍服务API,该API通常情况下不会因为版本更迭而失效.在对无障碍事件进行分析的时候,需要借助目标应用运行过程中产生的包名、类名以及控件ID.根据统计可知,应用在维护和升级时,一般不会对上述命名进行改动.

我们对SDK版本号进行了仔细的设计.CompileSdk使用最新的31,面向Android 12;通过指定targetSdkVersion 29,minSdk为23,为Android提供向前兼容.并且通过运行时检查系统版本的方式进一步解决兼容性问题.表2列出了Android 11、12上的兼容性设计.

表2　Android版本兼容性设计

Android版本	权限变化
12	请求ACCESS_FINE_LOCATION运行时权限,处理用户授予应用精确位置访问权限
11	请求ACCESS_BACKGROUND_LOCATION权限,在后台获取位置信息

为了维护用户的隐私,首先,我们的方法不会主动监控用户在应用上的行为.其次,提供一个应用选择功能,供用户选择目标应用.最后,供用户选择监控的粒度.以用户发送文本信息为例,在粗略的粒度下仅仅记录用户在什么时候发送过信息;在精细的粒度下会将用户的发送对象、信息内容等使细节记录下来.

4　基于无障碍服务的用户行为监控框架

通过上文的分析,监控方法的设计框架如图2所示.出于安全考虑,本文从权限、目标应用、监控粒度三个方面来保护用户的隐私.首先,本文仅仅需要申请有关网络、GPS(Global Positioning System)和无障碍服务的权限.其中,无障碍服务权限需要用户手动打开.只有用户主动打开了无障碍服务功能,监控才能运转.其次,用户自主选择待监控的应用.只处理包名在目标应用列表中的无障碍事件.最后,由用户选择监控粒度的大小.

当进行无障碍事件分发时,在尽可能保证响应界面所有变化的基础上,对无障碍事件进行过滤和延迟处理.然后,对无障碍事件进行分析和用户行为判定.从界面组件元素、无障碍事件的类型、界面焦点对象三个角度推断出用户的行为信息,并将关键数据记录在数据库中.采取SQLite数据库的存储方式.如图3所

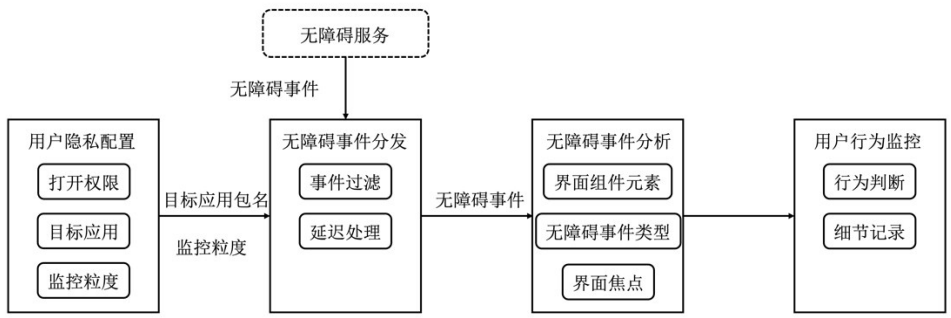


图2　基于无障碍服务的用户行为监控框架

示,一共创建4张表格,分别记录用户拍照活动、录音活动、发送消息活动、操作文件活动. 表格设计如图3所示. 使用文件传输协议(File Transfer Protocol,FTP),定期将这些数据加密后上传到指定的服务器上.

图3 数据表设计

4.1 无障碍事件分发

用户的一项动作往往会触发多种无障碍事件,为了提高性能,本文从无障碍事件过滤和无障碍事件延迟处理两个方面对无障碍事件进行分发.

如算法1所示,通过无障碍事件的getPackageName()获取根节点的包名. 将这些包名和数据库中记录的包名进行比较. 对于数据库中不存在的包名,直接丢弃对应的无障碍事件. 接下来通过延迟的方式,对多个无障碍事件一起进行处理. 在3.1小节,我们以系统应用(文件管理助手、相机、录音)以及市场应用QQ、微博、Gmail为例,对用户行为进行分析. 下面,我们同样以上述应用为例,对无障碍事件进行分析. 如表3所示,是启动这4类应用对应的无障碍事件的包名.

4.2 无障碍事件提取和分析

对无障碍事件进行提取和分析,推断其是否符合3.1节判定的用户敏感行为.

对于用户是否将隐私以文字方式进行记录和发送,需要关注的控件类型是EditText. 通过遍历无障碍事件根节点,判断当前节点是否是EditText类型. 对于含有EditText类型的节点的无障碍事件,说明当前界面上存在用户可以输入的文本编辑框. 用户在输入文字的时候,屏幕焦点会集中在文本编辑框上,对于EditText类型的节点,进一步通过无障碍事件的isFocused()判断当前编辑框是否被聚焦. 为了保护用户隐私,我们不会对用户输入的密码进行监控,在读取文本信息之前,使用无障碍事件的isPassword()方法,判断当前编辑框是否是密码框. 只有对于用户聚焦的、不是密码框的文本编辑框,才会通过无障碍事件的getText()方法获得输入框中的文本信息.

对于用户是否启动拍照活动,无论是启动系统自带的拍照活动,还是启动应用自带的拍照活动,都会出现无障碍事件,这些无障碍事件就归属这个活动所对应的类. 如表4所示,是我们记录得到的两个设备上的三个应用启动拍照活动所对应的包名和类名,其他应用并没有拍照功能. 一旦无障碍事件的根节点所对应的类属于表格中的类之一,说明用户启动了拍照活动,我们要将信息记录在数据库中. 根据拍照时间检索系统的图像和视频数据库,可以获得图片的详细属性信息.

算法1 无障碍事件分发框架

输入:无障碍事件arg0

输出:过滤后的无障碍事件arg0′

```
1.  packageName = arg0.getPackageName()
2.  if getDatabase.contains(packageName) then
3.      sendMessage(arg0, time_delay)
4.  end if
```

表3 启动应用对应的包名

应用名称	包名
Galaxy S10系统应用	com.sec.android.app.xx
Mi 10系统应用	com.android.xx
QQ	com.tencent.mobileqq
Gmail	com.google.android.gm
微博	com.sina.weibo

表4 拍照活动对应的包名和类名

应用名称	包名	类名
Galaxy S10系统拍照	com.sec.android.app.camera	com.sec.android.app.camera.Camera
Mi 10系统拍照	com.android.camera	com.android.camera.Camera
QQ	com.tencent.mobileqq	com.tencent.aelight.camera.aebase.QIMCameraCaptureActivity
微博	com.sina.weibo	com.sina.weibo.mediaproducer.activity.StoryCameraIndependentActivity

同理,对于用户是否启动录音活动,无论是启动系统的录音功能,还是启动应用的录音功能,都会出现无障碍事件,这些无障碍事件就归属这个活动所对应的类. 以Galaxy S10为例,启动系统自带录音功能会产生类名为com. sec. android. app. voicenote. main的无障碍事件. 另一种方式是判断无障碍事件中是否出现与录音功能有关的控件. 例如,QQ的录音功能不会触发具有明显特征类名的无障碍事件. 但是在QQ的录音界面

中,存在一个ID为press_to_speak_iv的录音控件. 我们只需要根据无障碍节点进行检索就可以判断当前用户是否在录音.

当用户使用应用界面,将存储在本地的文件/图片发送出去的时候,会存在选择文件/图片的过程. 应用一般调用Android系统提供的文件/图片选择接口,或者自行实现文件/图片选择器功能. 本文监控的应用中,只有QQ、Gmail和微博有选择并发送文件/图片的功能. 如表5所示,是记录得到的这三个应用打开图片/文件选择器时所对应的包名和类名. 一旦无障碍事件的根节点所对应的类名属于表格中类之一,说明用户开始选择文件/图片.

表5　发送文件对应的包名和类名

应用名称	包名	类名
Gmail	com.android.documentsui	com.android.documentsui.picker.PickActivity
QQ	com.tencent.mobileqq	tencent.mobileqq.filemanager.activity.FMActivity
微博	com.sina.weibo	com.sina.weibo.photoalbum.photoproducer

如表6所示,是记录得到用户使用系统自带的文件管理器对文件本身进行操作时所对应的包名. 当用户在对文件进行操作前,势必会有目录的进出操作,通过无障碍服务提供的findViewById(),找到目录控件对应的无障碍节点,通过getText()函数获取目录名称. 用户对文件的增删改查等一系列操作,系统往往会弹出对话框进行确认. 以删除文件为例,Galaxy S10会弹出对话框询问“是否将文件移至回收站”. 通过无障碍事件监控这些界面变化,获取系统信息,进而确定用户的操作.

表6　文件操作对应的包名和类名

应用名称	包名	类名
Galaxy S10文件管理器	com.sec.android.app.myfiles	com.sec.android.app.myfiles.external.ui. MainActivity
Mi 10文件管理器	com.android.fileexplorer	com.android.fileexplorer.FileExplorerTabActivity

根据表1,分析无障碍事件,鉴定当前用户行为是否属于敏感行为. 对于敏感行为,将应用名称、用户活动、以及细节信息,包括输入、浏览的文字、图片、语音、文件信息记录在数据库中供查询.

5　实验与结果分析

5.1　行为监控效果

5.1.1　实验配置

为了验证监控方法的可行性,分别在Galary S10和Mi 10上进行了测试,软硬件环境如表7所示. 在两台设备上,模拟现实生活中的场景,在四类应用上进行浏览、操作. 用户的行为将被记录在数据库中.

表7　手机软硬件环境

		Galaxy S10	Mi 10
硬件环境	处理器	骁龙855	骁龙 865
	内存	8 GB	8 GB
	外存	128 GB	256 GB
	电池容量	3 400 mAh	4 780 mAh
软件环境	操作系统	Android 12	Android 11

5.1.2　实验结果

表8是本文方法在四种应用上的用户行为监控结果. 实验证明本方法能100%监控用户是否使用目标应用. 对结果分析可知,用户运行应用时一定会触发无障碍事件,而它的包名正是该应用的包名. 对目标应用进行操作时,操作的背后对应类的方法的调用,通过进一步分析无障碍事件的类名,可以推断出用户的操作类别和使用的具体功能.

表8　本文方法在两台设备上,对于不同应用的用户行为监控结果

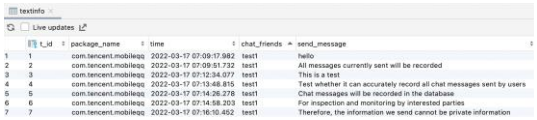
设备	应用	用户行为				
		拍照	录音	发送文字消息	发送图片/文件	文本增删改查
Galary S10	系统应用	✓	✓			✓
	QQ	✓	✓	✓	✓	
	Gmail	✓		✓	✓	
	微博	✓		✓	✓	
Mi 10	系统应用	✓	✓			✓
	QQ	✓	✓	✓	✓	
	Gmail	✓		✓	✓	
	微博	✓		✓	✓	

本方法能100%记录用户发送的文字信息,因为这些信息都是直观呈现在屏幕上的. 通过对基于视图层次结构编写的无障碍事件进行遍历,能获取这些信息,效果图如图4所示. 在图4(a)中,用户在手机应用中发送文本信息. 在图4(b)中,监控方法将应用名称、发送时间、发送对象、信息内容记录在数据库中. 但是,对于照片、语音和视频,我们仅仅能够判断用户是否产生、发送了这些多媒体信息,却无法获取它的详细属性,如文件路径,因为这些信息通常不会直观呈现在界面上.

5.2　性能影响和分析

作为一款监控用户行为的应用,本方法在后台静

(*a*) 用户在QQ上发送文本信息



(*b*) 监控应用的数据库

图4 QQ监控效果

默运行,不能影响用户正常使用手机. 下面,我们对本方法的延迟和资源开销进行分析.

5.2.1 延迟影响分析

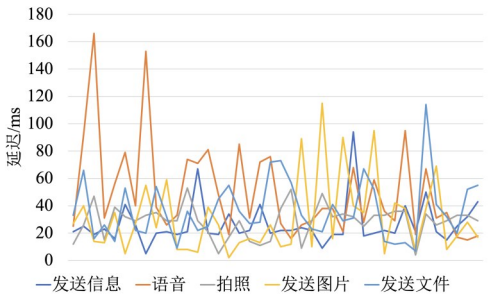
当出现无障碍事件的时候,系统回调无障碍事件处理函数. 影响系统回调的因素有很多,如处理器频率、单位时间内产生的事件数目等等. 它具有随机性、不确定性. 在这里,为了排除系统延迟带来的影响,测量从无障碍服务回调到完成用户行为记录需要的时间来估算本方法的延迟. 我们分别在Galaxy S10设备和Mi 10设备上以QQ这款应用为例,测试了200次用户操作,包括40次发送信息、40次语音、40次进行拍照、40次发送图片、40次发送文件操作. 结果如图5(*a*)所示,在Galaxy S10上最大延迟为166 ms,平均延迟为34.64 ms;如图5(*b*)所示,在Mi 10上,最大延迟为84 ms,平均延迟为24.19 ms. 延迟时间为毫秒级,可以忽略不计,因此几乎不会影响手机用户正常使用体验.

5.2.2 系统开销分析

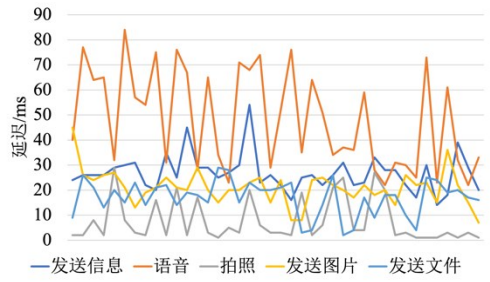
使用Android Studio提供的性能分析工具以及adb shell命令,对本文方法的CPU占用率、内存使用和能耗进行分析.

为了证明监控方法不会影响用户的手机使用体验,我们以热门应用网易云音乐后台播放作为对比,比较监控应用和网易云音乐后台播放时CPU占用率和内存使用. 具体过程如下,在前台模拟用户行为,一直运行目标应用QQ 30 min;启动监控应用,同样在前台使用QQ 30 min;关闭监控应用,启动网易云音乐的后台播放服务,保持在前台使用QQ 30 min. 分别记录三种情况下的CPU占用率和内存使用.

对于Galaxy S10,如图6所示,分别是三种情况下CPU使用率以及内存使用情况. 仅仅使用目标应用时,CPU平均占用率为34.41%,内存消耗为4 474.82 MB;使用目标应用的同时,开启监控应用,CPU占用率为42.46%,

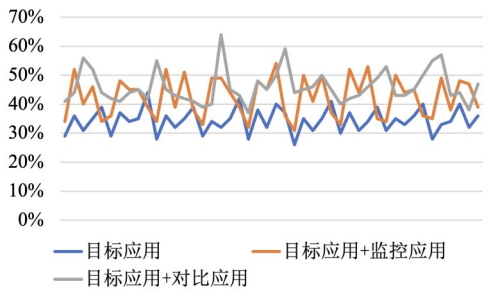


(*a*) Galaxy S10

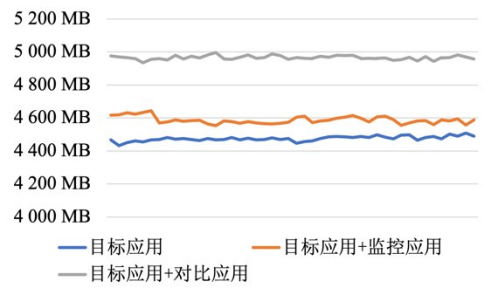


(*b*) Mi 10

图5 以QQ应用为例估算延迟



(*a*) Galaxy S10上CPU使用率



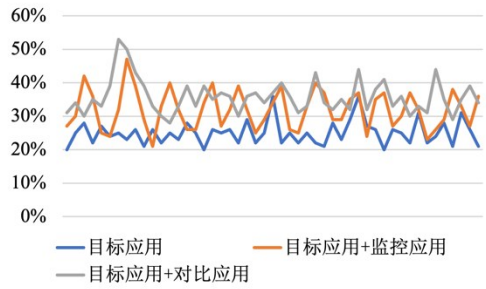
(*b*) Galaxy S10上内存使用情况

图6 Galaxy S10设备上的系统资源开销

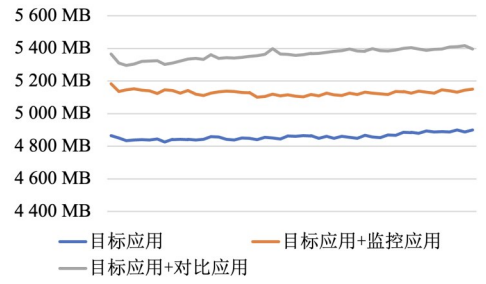
内存消耗为4 587.23 MB;使用目标应用,后台播放音乐,CPU占用率为45.98%,内存消耗为4 965.29 MB. 监控方法占据的CPU和消耗的内存明显小于网易云后台播放. 网易云后台播放作为常见的后台应用,在实际中不影响用户手机使用体验. 由此可得,监控应用同样不会

影响用户正常使用手机. 探查器将本方法的能源使用分类为light,因为它只使用了少量的CPU资源,没有使用任何位置资源、唤醒锁或警报.

对于Mi 10,如图7所示,分别是仅使用目标应用QQ、使用目标应用QQ和监控应用、使用目标应用QQ和对比应用网易云音乐的后台播放服务的CPU占用率、内存使用情况. 仅仅使用目标应用时,CPU平均占用率为24.98%,内存消耗为4 858.59 MB;使用目标应用的同时,开启监控应用时,CPU占用率为31.94%,内存消耗为5 128.60 MB;使用目标应用,同时后台播放音乐时,CPU占用率为35.80%,内存消耗为5 363.15 MB. 可见,我们的应用不会影响用户正常使用手机的体验. 同样,探查器将本方法的能耗使用分类为light,因为它只使用了少量的CPU资源,没有使用任何位置资源、唤醒锁或警报.



(*a*) Mi 10上CPU使用率



(*b*) Mi 10上内存使用情况

图7　Mi 10设备上的系统资源开销

基于上述分析得出结论,本方法对物理设备上的CPU、内存和能耗影响较小,可以作为后台应用程序持续监控用户行为.

6　结论及未来工作

尽管现有的研究中有很多关于恶意应用行为监控的方法,但如何有效监控用户行为仍然是一个挑战. 很多方法因为依赖系统版本和应用版本,只能固化地针对某种特定版本. 我们的工作展示了一种基于无障碍服务的用户行为监控的新方法,该方法适用于任何手机厂商的机型,不会因为版本更迭而失效. 通过实验验证它在两台设备上,针对四种不同的目标应用,正确记录用户行为. 通过性能分析,证明它可以实际部署在真实设备上,作为后台应用完成用户行为监控功能.

本方法具有较高的实际应用价值. 比如,在一个关注隐私的场所,将基于本文方法的监控工具装备在员工手机上,管理员就可以监控用户的行为,判定他们是否存在潜在的泄露保密信息的风险. 为了保障用户的隐私,本文方法的监控粒度也可以进行调整,从粗粒度的监控用户是否使用了某种应用,到细粒度的监控用户的使用细节. 本文仅仅通过四种应用验证方法的可行性. 实际上,对于市面上很多应用,本方法都能有效进行监控.

应用程序的视觉界面可能会随着版本的更新而变化,本方法的监控代码需要人为进行更新才能正确识别用户行为. 未来工作包括创建一个自适应框架,使其能够对应用程序新版本的布局参数进行自我学习.

参考文献

[1] Statista. Number of available applications in the Google Play Store from December 2009 to December 2021[EB/OL]. (2022). https://www. statista. com/statistics/266210/number-of-available-applications-in-the-google-play-store/.

[2] 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. (in Chinese)

[3] 何远, 张玉清, 张光华 . 基于黑盒遗传算法的Android驱动漏洞挖掘[J]. 计算机学报, 2017, 40(5): 1031-1043.
HE Y, ZHANG Y Q, ZHANG G H. Android driver vulnerability discovery based on black-box genetic algorithm. Chinese Journal of Computers, 2017, 40(5): 1031-1043. (in Chinese)

[4] 董超, 杨超, 马建峰, 等 . Android系统中第三方登录漏洞与解决方案[J]. 计算机学报, 2016, 39(3): 582-594.
DONG C, YANG C, MA J F, et al. The vulnerabilities and solutions of third-party login services in android system. Chinese Journal of Computers, 2016, 39(3): 582-594. (in Chinese)

[5] 马凯, 郭山清 . 面向Android生态系统中的第三方SDK安全性分析[J]. 软件学报, 2018, 29(5): 1379-1391.
MA K, GUO S Q. Security analysis of the third-party SDKs in the android ecosystem[J]. Journal of Software, 2018, 29(5): 1379-1391. (in Chinese)

[6] 张磊, 杨哲慜, 李明琪, 等 . TipTracer: 基于安全提示的安卓应用通用漏洞检测框架[J]. 计算机研究与发展, 2019,

56(11): 2315-2329.
ZHANG L, YANG Z M, LI M Q, et al. TipTracer: Detecting android application aulnerabilities based on the compliance with security guidance[J]. Journal of Computer Research and Development, 2019, 56(11): 2315-2329. (in Chinese)

[7] 李鹏伟, 姜宇谦, 薛飞扬, 等. 一种基于深度学习的强对抗性Android恶意代码检测方法[J]. 电子学报, 2020, 48(8): 1502-1508.
LI P W, JIANG Y Q, XUE F Y, et al. A robust approach for android malware detection based on deep learning[J]. Acta Electronica Sinica, 2020, 48(8): 1502-1508. (in Chinese)

[8] SATO R, CHIBA D, GOTO S. Detecting android malware by analyzing manifest files[J]. Proceedings of the Asia-Pacific Advanced Network, 2013, 36: 23-31.

[9] 郭春, 罗迪, 申国伟, 等. 一种基于诱导机制的间谍软件检测方法[J]. 电子学报, 2022, 50(4): 1014-1024.
GUO C, LUO D, SHEN G W, et al. A spyware detection method based on inducement mechanism[J]. Acta Electronica Sinica, 2022, 50(4): 1014-1024. (in Chinese)

[10] 陈铁明, 杨益敏, 陈波. Maldetect: 基于Dalvik指令抽象的Android恶意代码检测系统[J]. 计算机研究与发展, 2016, 53(10): 2299-2306.
CHEN T M, YANG Y M, CHEN B. Maldetect: An android malware detection system based on abstraction of Dalvik instructions[J]. Journal of Computer Research and Development, 2016, 53(10): 2299-2306. (in Chinese)

[11] SHABTAI A, KANONOV U, ELOVICI Y, et al. "Andromaly": A behavioral malware detection framework for android devices[J]. Journal of Intelligent Information Systems, 2012, 38(1): 161-190.

[12] ENCK W, GILBERT P, HAN S, et al. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones[J]. ACM Transactions on Computer Systems, 2014, 32(2): 1-29.

[13] 陈长青, 郭春, 崔允贺, 等. 基于API短序列的勒索软件早期检测方法[J]. 电子学报, 2021, 49(3): 586-595.
CHEN C Q, CUO C, CUI Y H, et al. Ransomware early detection method based on short API sequence[J]. Acta Electronica Sinica, 2021, 49(3): 586-595. (in Chinese)

[14] Developers Google . Create your own accessibility service [EB/OL]. (2022-02-23)[2022-03-26]. https://developer.android.com/guide/topics/ui/accessibility/service.

[15] Developers Google. Testing UI for multiple apps[EB/OL]. (2022-03-18)[2022-03-26]. https://developer.android.com/training/testing/ui-testing/uiautomator-testing.html.

[16] MA X X, YAN B, CHEN G L, et al. Design and implementation of a toolkit for usability testing of mobile apps [J]. Mobile Networks and Applications, 2013, 18(1): 81-97.

[17] PATERNÒ F, SCHIAVONE A G, CONTI A. Customizable automatic detection of bad usability smells in mobile accessed web applications[C]//Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services. New York: ACM, 2017: 1-11.

[18] DEKA B, HUANG Z F, KUMAR R. ERICA: Interaction mining mobile apps[C]//Proceedings of the 29th Annual Symposium on User Interface Software and Technology. New York: ACM, 2016: 767-776.

[19] LETTNER F, HOLZMANN C. Automated and unsupervised user interaction logging as basis for usability evaluation of mobile applications[C]//Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia. New York: ACM, 2012: 118-127.

[20] COSTAMAGNA V, ZHENG C. ARTDroid: A virtual-method hooking framework on android ART runtime[EB/OL]. (2022-01-01) [2022-03-26]. https://ceur-ws.org/Vol-1575/paper_10.pdf.

[21] TAN Y A, FENG S, CHENG X C, et al. An android inline hooking framework for the securing transmitted data [J]. Sensors (Basel, Switzerland), 2020, 20(15): 4201.

[22] Lody. VirtualApp[EB/OL]. (2021-11-03)[2022-03-17]. https://github.com/asLody/VirtualApp/blob/master/doc/VADev.md.

[23] 李晓娟, 陈海波. 基于分布式信息流控制的无障碍辅助性服务安全加固[J]. 软件学报, 2018, 29(5): 1318-1332.
LI X J, CHEN H B. Security reinforcement of accessibility service based on decentralized information flow control [J]. Journal of Software, 2018, 29(5): 1318-1332. (in Chinese)

[24] KALYSCH A, BOVE D, MÜLLER T. How android's UI security is undermined by accessibility[C]//Proceedings of the 2nd Reversing and Offensive-oriented Trends Symposium. New York: ACM, 2018: 1-10.

[25] DIAO W, ZHANG Y, ZHANG L, et al. Kindness is a risky business: On the usage of the accessibility {APIs} in Android[C]//22nd International Symposium on Research in Attacks, Intrusions and Defenses. Berkeley: USENIX, 2019: 261-275.

[26] Google. Get started on android with TalkBack - android

accessibility help[EB/OL]. (2020). https://support.google.com/accessibility/android/answer/6283677?hl=en.

[27] ZHANG X Y, DE GREEF L, SWEARNGIN A, et al. Screen recognition: Creating accessibility metadata for mobile applications from pixels[C]//Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. New York: ACM, 2021: 1-15.

[28] SUN H T, JIN C J, HELU X H, et al. Research on android infiltration technology based on the silent installation of an accessibility service[J]. International Journal of Distributed Sensor Networks, 2020, 16(2): 1550147720903628.

[29] BORGES N P, RAU J, ZELLER A. Speeding up GUI testing by on-device test generation[C]//Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering. New York: ACM, 2020: 1340-1343.

[30] NEGARA S, ESFAHANI N, BUSE R. Practical android test recording with espresso test recorder[C]//2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice. Piscataway: IEEE, 2019: 193-202.

[31] XIE N, NI Y R, LIU X X, et al. Implementation of simulation control automation tool based on android accessibility service[J]. Journal of Physics: Conference Series, 2021, 1881(3): 032071.

[32] YANG Z M, YANG M, ZHANG Y, et al. AppIntent: Analyzing sensitive data transmission in android for privacy leakage detection[C]//Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. New York: ACM, 2013: 1043-1054.

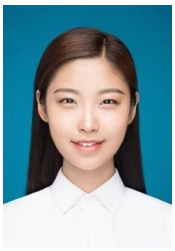
[33] PETITTI J. Appjudicator: Enhancing Android Network Analysis Through UI Monitoring[D]. Worcester: Worcester Polytechnic Institute, 2021.

[34] HUANG J, BACKES M, BUGIEL S. A11y and Privacy don't have to be mutually exclusive: Constraining accessibility service misuse on Android[C]//Proceedings of the 30th USENIX Security Symposium. Berkeley: USENIX, 2021: 3631-3648.

作者简介

马文聪　女，1999年出生于安徽省池州市. 2021年于北京理工大学获得学士学位. 现为北京理工大学计算机学院硕士研究生. 主要研究方向为信息安全.
E-mail: 3120211066@bit.edu.cn

谭毓安　男，1972年1月出生于重庆巫溪. 北京理工大学网络空间安全学院教授. 主要研究方向为人工智能安全、系统安全.
E-mail: tan2008@bit.edu.cn



冯　硕　女，1997年出生于河北省唐山市. 2019年于河北工业大学获得学士学位，2022年于北京理工大学获得硕士学位. 主要研究方向为信息安全.
E-mail: 17330582191@163.com

刘　璐　女，1992年出生于四川省广安市. 2017年于北京理工大学获得硕士学位. 现为北京理工大学计算机学院实验师. 主要研究方向为信息安全与机器学习.
E-mail: liulu@bit.edu.cn

李元章（通讯作者）　男，1978年2月出生于江苏盐城. 分别于2001、2004及2015年获得北京理工大学学士、硕士和博士学位. 现为北京理工大学计算机学院副教授. 主要研究方向为信息系统安全、人工智能安全等.
E-mail: popular@bit.edu.cn