基于 Android 权限机制的动态隐私保护模型*

戴 威, 郑 滔
(南京大学 软件学院, 南京 210093)

摘 要: 针对 Android 平台自身粗粒度权限机制的缺陷以及缺乏有效预防程序间隐私泄露机制的问题,提出一种改进的细粒度权限配置机制与隐私数据动态着色隔离相结合的 Android 隐私保护模型。通过对 Android 应用程序权限进行细粒度的动态配置,阻断隐私数据从程序内部泄露的途径,利用隐私数据着色跟踪实现对程序间传播的包含不同隐私权限标签的消息的隔离控制。通过大量实验的反复测试,该模型可以有效保护 Android 程序内部的隐私数据,及时发现程序间权限提升攻击进而实现隐私数据隔离,从而全方位实现 Android 隐私数据的保护,并为以后相关研究提供了新的方向。

关键词: Android 平台; 隐私保护; 权限提升攻击; 细粒度权限; 动态着色; 组件间通信
中图分类号: TP309 **文献标志码**: A **文章编号**: 1001-3695(2012)09-3478-05
doi:10.3969/j.issn.1001-3695.2012.09.074

Dynamic privacy protection model based on Android permissions

DAI Wei, ZHENG Tao
(*Software Institute*, *Nanjing University*, *Nanjing* 210093, *China*)

Abstract: The privacy mode in Android has some defects and is lack of effective mechanisms to prevent loss of privacy between applications. This paper proposed a new privacy protection model by using the fine-grained permissions and privacy data tainting. Firstly the method could prevent loss of privacy from a single application by dynamically configuring the permissions in a fine-grained manner. By leveraging the track of the tainted privacy data it could prohibit the propagation of the privacy data between applications which had different permissions on the accordingly privacy tag. Repeatly experimental tests show the model is able to effectively protect the privacy of data within the Android application, and to detect the privilege escalation attacks between applications so as to isolate the private data. As a result, the full range of private data within Android is in protection, and leads a new direction for related research in the future.

Key words: Android platform; privacy protection; privilege escalation attack; fine-grained permissions; dynamic tainting; inter-component communication

0 引言

智能手机的使用越来越广泛。最新一期的 Gartner 报告[1]指出,2011 年第三季度全球手机销售达到 4.405 亿,其中 Android 达到智能手机市场 52.5% 的份额,较去年增加了一倍,占据智能手机操作系统的第一位。越来越多的用户习惯于利用手机来储存和处理私人及商业信息。

最新研究[2]指出,针对 Android 平台的攻击呈现复杂多变的趋势。分析表明,大部分的攻击手段着眼于隐私层面,滥用程序权限和串谋权限攻击引起的隐私泄露案例越来越多,Soundcomber[3]就是一个典型的攻击案例。针对 Android 平台的隐私安全问题,主流的研究方向集中在静态分析和动态控制两个方面。Enck 等人[4]引入一组静态的权限策略,对违反权限策略的应用拒绝安装,缺点是需要考虑所有可能的情况,不能有效预防权限提升攻击。Felt 等人[5]静态分析应用是否满足程序功能所需的最小权限集并向用户提供安装建议,缺点是误报率高。在动态研究方面,Enck 等人[6]对隐私数据着色追踪,当隐私数据通过网络接口离开系统时向用户发送警告,缺点是 TaintDroid 项目只是作为隐私数据流监视的工具,并没有提供有效防控隐私泄露的机制。但是基于 TaintDroid 平台的隐私防控机制成为近年研究的热点,本文项目同样在此基础上作了进一步防控拓展。TISSA 项目[7]是近期在 Android Market 发布的隐私保护程序,该项目向用户提供了修改程序隐私相关权限的功能以期提供对应用细粒度的权限控制机制,较早的一些类似应用仅仅实现了禁止某些隐私权限,却容易造成程序崩溃,相比之下,该方案提供了多粒度的选择,但是实验表明,针对多个程序组合串谋攻击造成的隐私泄露 TISSA 没有丝毫办法。

针对传统研究方法的问题,本文提出一种细粒度权限配置和数据动态着色跟踪隔离相结合的隐私保护模型 CrossDroid。

1 Android 基本概念

1.1 Android 架构

Android 是一个开放的移动平台框架,主要包括 Linux 内

收稿日期: 2012-02-05; **修回日期**: 2012-03-14 **基金项目**: 国家自然科学基金资助项目(61105069);国家“863”计划资助项目(2007AA01Z448)

作者简介:戴威(1986-),男,江苏徐州人,硕士,主要研究方向为程序安全(MG0932002@software.nju.edu.cn);郑滔(1966-),男,教授,主要研究方向为程序安全.

核层、中间件层和应用层,如图 1 所示。其中 Linux 内核层提供设备驱动和底层服务。中间件层由本地库、Android 运行时环境和应用框架层组成。Android 运行时环境包括一个优化的 Java 虚拟机 Dalvik virtual machine(DVM)和核心 Java 库。DVM 用于解释执行应用可执行文件。应用框架层由提供系统服务的系统程序组成,其中 activity manager 负责管理应用的生命周期。本文提出的 CrossDroid 提供基于中间件的策略模型。Android 应用层包含了 Android 提供的核心应用并支持第三方软件。Android 应用主要包括 activities、services、content providers 和 broadcast receivers 四种组件。

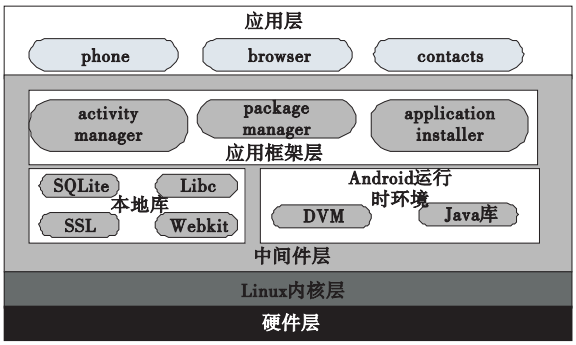

图1　Android系统架构

1.2　Android 进程间通信机制

Android 提供多种程序间通信形式。如图 2 所示,Android 以一种基于 binder 的轻量级进程间通信模型 ICC(inter-component communication)作为组件间通信的主要形式。Android 同样支持标准的 Linux IPC 通信机制。此外,应用程序之间还可以利用 Internet socket 连接实现进程间通信。CrossDroid 主要关注 ICC 通信中的隐私保护机制,对 Linux IPC 及 socket 通信的隐私监控机制将在下一步实现。

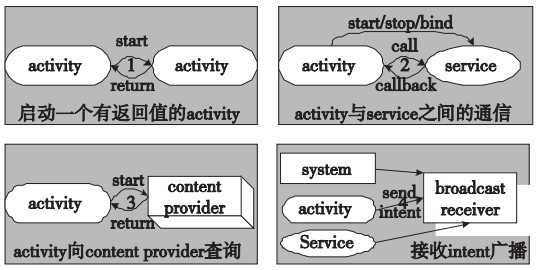

图2　Android进程间通信形式

1.3　Android 安全机制

Android 的核心安全机制主要包括程序沙箱、应用签名和权限框架。Android 的权限框架限制了应用程序对敏感数据(如短信、通讯录等)、资源(如电量和日志文件)及系统接口(如 Internet、GPS 等)的访问。程序所需权限列表在安装时必须被用户全部赋予,安装完成后不可更改,这种粗粒度的特性导致了潜在的安全缺陷。

一方面,赋予程序访问隐私数据权限的负担被转嫁到没有安全意识的用户头上,程序对授权隐私数据的使用不再受到控制,Android 也不会通知用户;另一方面,Android 缺乏应对权限提升攻击的有效机制[8]。研究最新的攻击手段表明,非授权应用可以利用其他程序的漏洞间接获得隐私数据;攻击者还可以组合多个已感染的或恶意程序形成串谋攻击[3],利用组合权限窃取隐私。TISSA[7]应用实现了 Android 细粒度的权限机制,但缺乏有效应对串谋攻击的手段。

2　CrossDroid 模型

总的来说,CrossDroid 隐私保护模型在 Android 权限框架基础上提供更细粒度的控制手段(用户可以对程序的每个隐私权限项进行配置),并利用隐私权限标记对隐私数据动态着色跟踪防止程序间消息通信引起的隐私泄露。

2.1　CrossDroid 基本架构

CrossDroid 主要包括权限设置模块、隐私数据访问模块、隐私数据着色模块和 ICC 数据监控模块,如图 3 所示。

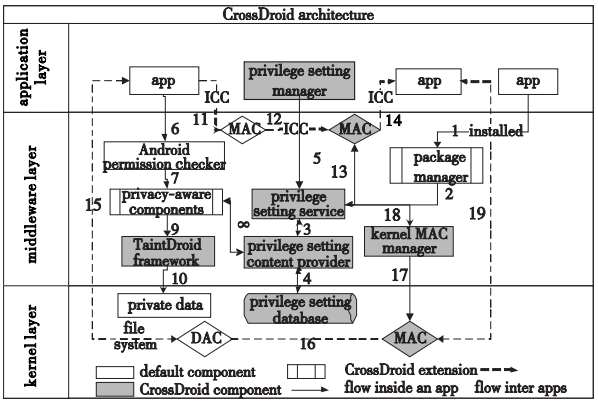

图3　CrossDroid隐私保护模型架构

权限设置模块向用户提供对应用程序细粒度的权限设置。包括一个权限设置管理软件 privilege setting manager、package manager 和 application installer 的权限扩展插槽、提供权限策略配置存储的 privilege setting content provider。

隐私数据访问模块包括隐私数据源的权限检查扩展插槽,具体指在隐私内容的提供者和服务模块添加的扩展槽。

权限设置模块和隐私数据访问模块与 TISSA[7]的设计类似,但是在提供隐私权限选项及策略上 CrossDroid 基于 TaintDroid 的着色标签类型提供更细粒度的对应关系,并提供更加简单和类别化的策略定制推荐。

隐私数据着色模块基于 TaintDroid 项目。TaintDroid[6]在中间件级别对隐私数据进行着色实现隐私数据监视。CrossDroid 基于 TaintDroid 项目的着色类型与权限设置模块的细粒度权限标签进行对应采取监控策略。

ICC 数据监控模块主要包括提供应用权限策略配置查询服务的 privilege setting service 以及 Android 强制访问控制模块(MAC)扩展插槽。MAC 扩展插槽在程序间通信发生数据交换时被触发,向 privilege setting service 查询程序之间相异权限是否与消息着色类型重叠进而执行相应策略。

2.2　CrossDroid 流程

CrossDroid 细粒度权限设置模块流程如图 4 所示,步骤 1)~4)描述新软件安装时,权限设置模块默认用户赋予其隐私相关权限并存储;步骤 5)~9)描述用户通过权限设置管理软件对程序的隐私权限项进行配置的过程。

图 5 详细描述了 CrossDroid 隐私数据访问模块的处理流程,步骤 1)~3)描述 Android 自有的权限检查过程;步骤 4)~10)描述请求转发到隐私源组件时,触发权限扩展插槽查询请求者的权限策略,策略禁止则返回给调用者伪造数据;策略允

许则经过 TaintDroid 处理返回着色隐私数据。

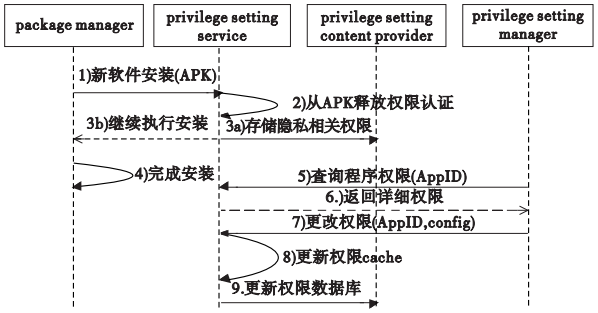

图4 CrossDroid细粒度权限设置流程

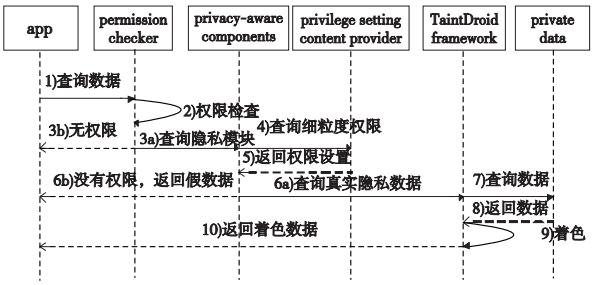

图5 CrossDroid隐私数据访问流程

针对图 5 的处理流程,CrossDroid 程序间 ICC 数据监控模块如图 6 所示(其他三种通信形式处理机制参见下一章实现描述)。Activity manager 中 reference monitor 执行步骤 1)~3)的常规权限检查后,触发 CrossDroid 的 MAC 扩展模块查询通信消息的着色类型集与程序之间的相异隐私权限集是否存在重叠,如果重叠则执行步骤 7)。例如,如果通信双方在权限 access_fine_location 配置上不同但消息包含 taint_location 标签,则触发 CrossDroid 禁止通信策略。

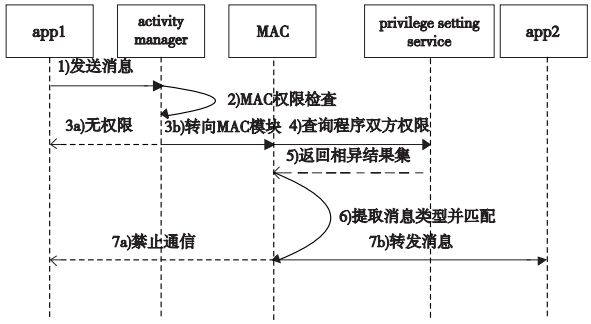

图6 CrossDroid程序间隐私保护流程

3 CrossDroid 模块实现

3.1 细粒度权限设置模块

在细粒度权限设置模块中,权限设置内容提供者 privilege setting content provider 管理存储程序权限策略的 SQLite 数据库,并向隐私设置服务 privilege setting service 和隐私源权限插槽提供查询调用者权限策略的接口。查询接口函数以程序名为参数,返回表示调用者当前隐私权限设置的 32 bit int 值,设置接口函数以程序名和更新的权限值为参数。权限策略数据库储存了应用权限策略的私有数据,为防止被恶意程序窜改,CrossDroid 采用签名机制使得该数据库内容只能被具有相同签名的权限设置管理软件所访问。

权限管理软件 privilege setting manager 向用户提供了当前第三方应用程序的权限策略配置,该软件主要由两个 activities 组成:privilege setting manager activity 向用户提供已安装软件的列表;app privilege setting activity 提供配置应用每一项隐私权限的界面。设计权限细粒度选项时,综合文献[7],为向用户提供更好的程序透明性并避免禁止权限而引起程序崩溃,仅提供给权限项 trusted、bogus 两种选择。此外权限管理软件提供了类别化的软件清单分类视图并提供推荐策略配置,实现了类别化的策略配置挖掘。

Package manager、package installer 的权限插槽负责第三方应用安装时对程序隐私权限集的默认初始化工作。

定义 1 系统隐私权限集。对应 TaintDroid 着色类型组合及 Android 提供的隐私权限列表生成的 CrossDroid 隐私权限配置集合 $P=\{p_1,p_2,\cdots,p_{32}\}$,其中 p_i 的对应权限部分片段如表 1 所示。

表 1 CrossDroid 隐私权限与 Android 权限对应表片段

分类	隐私权限 p_i	Android 对应授予权限 q_i
地理位置	ACCESS_COARSE_LOCATION	android. permission. ACCESS_FINE_LOCATION
信息	ACCESS_FINE_LOCATION	android. permission. ACCESS_COARSE_LOCATION
手机信息	READ_PHONE_STATE	android. permission. READ_PHONE_STATE
短信息	READ_SMS	android. permission. READ_SMS
通讯录	READ_CONTACTS	android. permission. READ_CONTACTS

定义 2 软件隐私权限集。给定系统隐私权限集 $P=\{p_1,p_2,\cdots,p_{32}\}$,对应的 Android 隐私权限列表集合 $Q=\{q_1,q_2,\cdots,q_{32}\}$,软件 A 权限集 $Q_a=\{a_1,a_2,\cdots,a_j\}$,$A$ 的隐私权限集定义为 $P_a=\{b_1,b_2,\cdots,b_k\}$,其中 $P_a\subseteq P$,对于 $\forall a_i\in Q_a$,如果 $a_i\in Q$,则 a_i 映射到 P 中的项 $p_k\in P_a$。

3.2 隐私数据访问模块

隐私数据访问模块核心由隐私源 privacy-aware component 的权限插槽组成。根据 CrossDroid 隐私数据的划分,隐私源主要包括 location manager、telephony manager、contacts content provider。权限扩展插槽负责查询 privilege setting content provider 的接口,返回真实或伪造的数据。

3.3 隐私数据着色模块

隐私数据着色模块基于隐私保护研究项目 TaintDroid[6]。TaintDroid 对位于中间件层的 Dalvik virtual machine 字节码解析部分修改实现着色数据处理,向隐私数据源添加 hook 函数对隐私数据进行着色。CrossDroid 利用 TaintDroid 的着色类型进一步权限整合。TaintDroid[6] 实现时修改了 DVM 虚拟机的数据处理指令源码,通过在内存栈里向隐私数据分配额外的 32 bit 空间储存着色标签;在隐私数据获取途径上利用 Android 提供的资源访问模型,在隐私数据源提供模块中加入 hook 函数调用着色函数进行着色。

图 7 描述了 TaintDroid 的架构设计。隐私信息在隐私源(如 contacts provider)获取数据时被标记(1);着色标记接口调用自己提供的本地方法(2)将着色信息储存在着色表里;当着色信息被用于 IPC 传播时,被修改的库(3)对传送的包 parcel 着色,在实现时,parcel 包通过内核的 IPC 机制进行传播(4)并被远程非信任程序接收;被修改的 binder 库(5)将接收到的 parcel 包提取着色信息并储存在自己的着色表里;当隐私数据被调用时(7),DVM 解释器负责对着色信息的处理。形成一个

完整的 TaintDroid 着色流程。

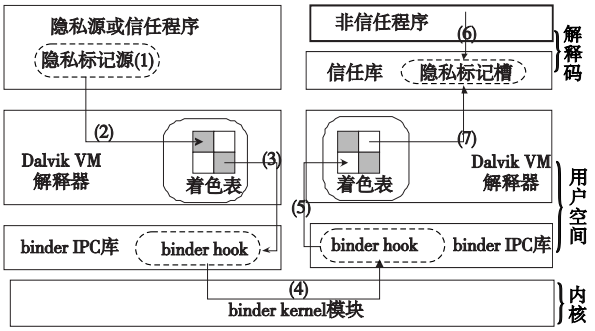

图7　TaintDroid架构

TaintDroid 利用图 7 的方式实现隐私数据在系统内的着色传递,当程序调用网络接口发送含有着色标签的数据包时,TaintDroid 记录并发送警告。

可以看出,TaintDroid 只是监视 Android 隐私数据从网络泄漏的工具,并没有提供阻止隐私泄露的措施,而且容易造成对刻意处理过的数据丢失跟踪和着色爆炸两种极端结果。CrossDroid 利用 TaintDroid 提供的着色机制在消息级别的着色传递环节实施防控策略,及时阻断隐私消息的泄露,降低着色爆炸的生成概率。

3.4　ICC 数据监控模块

组件间通信(ICC)是 Android 程序间通信主要形式。由图 2 可知,ICC 具体有四种形式:a) activity 之间的通信;b) activity 绑定到其他程序的 service;c) activity 查询其他程序的 content provider;d) 通过 broadcast 实现程序间的隐式通信。总的来说,可以分为直接的 ICC 和间接的 ICC 两种,以上列举除了 d) 属于间接 ICC 其他都是直接 ICC。

Android 采用 binder 来实现 ICC。Binder 是一种基于 client-server 的通信方式,通过匿名共享内存映射到不同进程的地址空间实现数据共享。传输过程中数据可以以 parcel 序列化格式传递。借助 TaintDroid 提供的针对 parcel 包的着色函数 updateTaint( )、getTaint( ),ICC 数据监控模块在 ICC 发生时被触发,调用 privilege setting service 检测 Android 消息级别的隐私泄露。

为防止直接 ICC 导致的隐私泄露,CrossDroid 封装了 activity manager 的 checkPermisson 函数,以双方 ID、消息为参数查询是否属于异色消息传递。

针对图 7 的情况,在 libbinder 模块中对向 content provider 查询返回的标准 cursor 调用 cursor window 提供的函数解压修改,过滤掉通信双方相异权限着色的数据包,阻断利用 content provider 实现串谋隐私泄露的通道。

定义 3　异色消息传递。给定通信双方程序 A、B 以及传递的消息 M,$P_a=\{a_1,a_2,\cdots,a_j\}$、$P_b=\{b_1,b_2,\cdots,b_k\}$、$P_m=\{m_1,m_2,\cdots,m_l\}$ 分别为 A、B、M 的隐私权限集,定义异色消息传递为 $((P_a-P_b)\cup(P_b-P_a))\cap P_m\neq\varnothing$。

针对图 6,为阻止异色消息传递在间接 ICC 中传递,向 activity manager broadcast 管理模块加入 hook 过滤异色消息传递的组合,形成细粒度权限层次上的程序域划分隔离。

4　实验评估

本次实验的配置为 VMware WorkStation 7.0 上 Ubuntu 11.10 虚拟机,2 CPU,2 GB RAM。Android 的源码版本是 Android 2.3.4_r1,内核源码是 Linux kernel 2.6.29 goldfish。攻击模型在编译生成的 Emulator 模拟器上进行。

4.1　测试用例设计

根据当前 Android 隐私泄露的基本形式,本次实验通过分析提供的多个攻击案例,设计了三类测试用例,如表 2 所示。

表 2　隐私泄露测试用例设计表

	1st 应用	2nd 应用
1	具有 ACCESS_FINE_LOCATION 和 Internet 权限的应用(游戏居多,例如 Paper Toss)	
2	具有 ACCESS_FINE_LOCATION 的恶意计步器软件	具有 Internet 权限的恶意墙纸软件
3	通过 CrossDroid 禁掉 Internet 功能的恶意导航仪软件	具有 Internet 权限的恶意墙纸软件
4	易受攻击的浏览器软件	具有 READ_SMS 权限的恶意短信管理器
5	具备访问私有数据权限且利用 service 后台监听数据变化的软件	恶意注册广播接收私有数据并具有 Internet 权限的软件

测试用例 1 代表当前大部分游戏软件的特征,以 Paper Toss 游戏为例,Internet 权限保证在线成绩排名和广告推送,利用获取的地理位置信息辅助 Mobi Ad 的显示。本次实验设置 Paper Toss 的 ACCESS_FINE_LOCATION 权限为 bogus 以提供虚假的地理信息。

测试用例 2、3 代表了最新的串谋权限攻击,由于 Android Market 的低门槛性,任何人只要交纳 25 美金后就可以上传自己的应用供别人下载,这就构成了多个恶意软件串通起来进行权限整合攻击的条件。借助 ICC 形式,恶意软件之间发送带有隐私内容的消息实现隐私泄露,CrossDroid 的 ICC 监控模块被用来测试监控的效果。

测试用例 4、5 利用了现有软件的缺陷,如提供不安全的 service 和没有权限设置的广播,容易被恶意的第三方软件所窃听挟持,ICC 监控模块和 broadcast 钩子函数负责测试捕捉的效果。

4.2　有效性分析

本次实验随机从 Android Market 上下载了 20 个应用,其中包括 5 个具有 Internet 权限的游戏、3 个导航软件、2 个通讯录管理软件、10 个实用软件(团购导航、天气预报、电影、娱乐信息导航)。Android CM 开源项目虽然引入了可以禁止应用某个权限的功能,却容易造成应用由于无法获得相应权限而引起的程序崩溃问题。因而在本次实验中 TISSA 和 CrossDroid 的隐私权限选项设计为 bogus(虚假的)。根据 TaintDroid 跟踪的结果表明(如表 3 所示),TISSA 和 CrossDroid 保护程序隐私数据成功且没有崩溃的数目相同,在程序内部隐私数据保护上 CrossDroid 达到了相同的效果。

表 3　TISSA 与 CrossDroid 应用隐私权限定制成功数

隐私类别	TISSA bogus 成功数	CrossDroid bogus 成功数
Internet	20	20
location	14	14
phone ID	11	11
contacts	2	2
call log	2	2

针对测试用例 2 ~ 5,采用了经典的 FriendTracker 和 FriendViewer 应用案例[9]以及自定义的简易攻击模型。

实验结果如表 4 所示,CrossDroid 捕捉到了全部着色消息,其中成功禁止异色消息传递率为 89.7%。由于 TaintDroid 数据着色传递中存在着色爆炸缺陷,导致 CrossDroid 检测到的着色消息的标记存在一定的误报率,但是实验数据表明,误报数在可承受范围内。同时,实验结果也表明,TISSA 对这种程序间通信引起的隐私数据泄露的串谋攻击手段没有任何有效措施。

表 4 TISSA 与 CrossDroid 应对串谋攻击实验结果

采用模型	着色消息数	消息捕捉数	禁止通信数	异色消息传递	误报数	成功率/%
CrossDroid	37	37	29	26	3	89.7
TISSA	26	0	0	22	0	0

4.3 性能测试

为评估 CrossDroid 的性能负载,本次实验使用 Java 性能评估工具 Caffeine Mark 3.0 评估 CrossDroid 和 Android 原始系统的性能。CM 3.0 包含一系列的程序集测量应用运行时负载。由于 CrossDroid 基于 Android 中间件层,CM 的结果能真实反映 CrossDroid 的性能差别。

通过比较图 8 可知,CrossDroid 相对于原始的系统分数差别并不明显。其中差距比较大的主要在 string、float 数据得分上,由于 CrossDroid 着色模块基于 TaintDroid 平台,而 TaintDroid 对原型数据处理作了较多的改动,差别较大在推断之中。

CM 的数据表明 CrossDroid 具有良好的性能支出,对原始系统的负担可忽略不计。

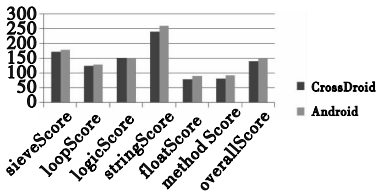

图8 Caffeine Mark 3.0测试结果

5 结束语

本文基于 Android 的应用权限机制提出一种细粒度的权限模型,并利用对隐私数据的动态着色将 Android 应用划分为不同的权限域,使 Android 隐私数据在程序内部和程序之间都得到准确高效的保护。对模型原型的评估证实该模型能够有效且全面地保护隐私数据。下一步的工作将考虑融合 Linux 内核层的安全机制,将该模型扩展到内核层级,以支持标准 Linux IPC、文件系统和 socket 连接上的隐私保护。

参考文献:

[1] Gartner november report [EB/OL]. [2011-11-15]. http://www.gartner.com/it/page.jsp?id=1848514.

[2] ENCK W, OCTEAU D, MCDANIEL P, *et al*. A study of android application security [C]//Proc of the 20th USENIX Security Symposium. Berkeley: USENIX Association, 2011.

[3] SCHLEGEL R, ZHANG K, ZHOU X, *et al*. Soundcomber: a stealthy and context-aware sound trojan for smartphones [C]//Proc of the 18th Annual Network and Distributed System Security Symposium. 2011: 17-33.

[4] ENCK W, ONGTANG M, MCDANIEL P. On lightweight mobile phone application certification [C]// Proc of the 16th ACM Conference on Computer and Communications Security. New York: ACM, 2009: 235-245.

[5] FELT A P, CHIN E, HANNA S, *et al*. Android permissions demystified [C]//Proc of the 18th ACM Conference on Computer and Communications Security. New York: ACM, 2011.

[6] ENCK W, GILBERT P, B G, *et al*. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones [C]// Proc of the 9th USENIX Symposium on Operating Systems Design and Implementation. Berkeley: USENIX Association, 2010.

[7] ZHOU Ya-jin, ZHANG Xin-wen, JIANG Xu-xian, *et al*. Taming information-stealing smartphone applications (on Android) [C]//Proc of the 4th International Conference on Trust and Trustworthy Computing. 2011: 93-107.

[8] DAVI L, DMITRIENKO A, SADEGHI A R, *et al*. Privilege escalation attacks on Android [C]//Proc of the 13th Information Security Conference. 2010: 346-360.

[9] ENCK W, ONGTANG M, McDANIEL P. Understanding android security[J]. IEEE Security & Privacy Magazine, 2009, 7(1): 50-57.