Android 设备中基于流量特征的隐私泄露评估方案

王竹[1,2]，贺坤[1,2]，王新宇[1,2]，牛犇[1]，李凤华[1,2]
（1. 中国科学院信息工程研究所，北京 100093；2. 中国科学院大学网络空间安全学院，北京 100049）

摘 要：针对 Android 操作系统 App 内第三方域名采集用户信息造成的隐私泄露问题，基于 TF-IDF 模型和层次聚类方法提出了移动设备中的隐私泄露评估方案 HostRisk。TF-IDF 模型通过 App 内域名的行为特征计算域名与 App 的业务相关性，对于未能表现出 App 业务相关性行为特征的业务相关域名通过平均连接的凝聚型层次聚类方法进行调整优化，最终根据 App 内所有域名的排名计算其隐私泄露危害程度。实验结果验证了所提方案的有效性和效率。

关键词：Android；隐私泄露；隐私评估；隐私保护
中图分类号：TN 929
文献标识码：A
doi:10.11959/j.issn.1000−436x.2020020

Traffic characteristic based privacy leakage assessment scheme for Android device

WANG Zhu[1,2], HE Kun[1,2], WANG Xinyu[1,2], NIU Ben[1], LI Fenghua[1,2]
1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: Aiming at the privacy leakage, which was caused by collecting user information by third-party host in Android operating system App, a privacy leakage evaluation scheme HostRisk was proposed. HostRisk was based on TF-IDF model and hierarchical clustering method, which was applied in mobile device. The TF-IDF model calculated the business relevance between Apps and hosts via the behavior characteristics of the hosts in these Apps. For the business related hosts that fail to express the business relevance characteristics, those hosts were adjusted and optimized via the average connected hierarchical agglomerative clustering method. Finally, the harmful degree of privacy leakage was evaluated based on the ranking of all hosts in the App. The experimental results verify the effectiveness and efficiency of the scheme.

Key words: Android, privacy leakage, privacy assessment, privacy preservation

1 引言

过去的 15 年内，智能手机的保有量呈现出爆炸式的增长态势，截至 2019 年 6 月，Android 以 79.90%的市场占有率成为中国移动终端操作系统市场的“领头羊”。为了满足用户的个性化需求，各类 App 层出不穷，其中免费 App 更是成为热门需求。Google Play 中提供多达 55 类 App，内容涵盖教育、生活、娱乐、健康等诸多方面，已经成为人们生活中不可或缺的一部分。

然而，在满足用户日常教育、生活、娱乐等需求的同时，用户隐私泄露问题日益突出[1]。基于权限管理的 Android 操作系统隐私泄露问题逐渐暴露，一些免费 App 的开发商更是通过在 App 中植

收稿日期：2019−09−17；**修回日期**：2019−12−06
通信作者：牛犇，niuben@iie.ac.cn
基金项目：国家重点研发计划基金资助项目（No.2017YFB0802203）；国家自然科学基金资助项目（No.61872441, No.61672515）；中国科学院青年创新促进会人才基金资助项目（No.2018196）
Foundation Items: The National Key Research and Development Program of China (No.2017YFB0802203), The National Natural Science Foundation of China (No.61872441, No.61672515), Youth Innovation Promotion Association CAS (No.2018196)

入非主业务第三方库的手段获取用户隐私信息从而谋取利润，从用户的角度考虑，用隐私信息交换 App 服务是不可接受的。近年来，App 盗用隐私信息造成用户生命财产安全受损的事件屡见不鲜。

已有研究成果表明[2~7]，大量 App 在运行时存在第三方域名收集用户隐私信息的行为。为此，许多研究[7-12]提出了相应的解决方案，解决这类问题的方法基本可以归纳为 2 类：1) 静态解析，通过反编译代码分析 App 结构、权限等特征检测恶意应用程序或收集用户隐私信息的第三方库，基于分组的依赖关系[8]和敏感应用程序接口（API, application program interface）调用统计特征[9]的方法是静态检测的流行做法；2) 动态解析，通过捕获 App 运行行为特征达到检测目的，基于第三方域名特征[10]和通信流量特征[11]是动态检测方案的有效手段。然而，已有的研究方案大都存在以下几个问题：1) 随着 App 的升级更新，新型恶意应用程序和非主业务第三方库不断涌现，传统检测方案效果和效率逐渐变差；2) App 中的第三方库检测和恶意应用程序检测无法得知用户的隐私信息发送给哪些第三方；3) App 级别的粗粒度检测方案不能满足检测 App 的每个数据分组泄露用户隐私信息的问题。

为了解决上述问题，本文提出了一种基于词频–逆文本频率（TF-IDF, term frequency-inverse document frequency）模型和层次聚类方法的隐私泄露评估方案 HostRisk。该方案通过捕获用户移动设备端中 App 的网络流量特征，基于 TF-IDF 模型计算 App 内域名的业务相关性，同时基于平均连接的凝聚型层次聚类方法优化未能表现出主业务相关性行为特征的 App 主业务域名的业务相关性得分，并根据 App 内的域名业务相关性排名表计算域名的隐私泄露程度，通过加权平均的方式评估 App 泄露用户隐私的风险。基于 TF-IDF 模型的业务相关性计算方法会根据域名的行为特征计算域名业务相关性，但存在部分主业务域名未能表现出与其相关的行为特征，例如与 App 不频繁交互的主业务域名，单独考虑行为特征并不能充分表现其业务相关的属性，进而使用基于平均连接的凝聚型层次聚类方法进行调整和优化。App 中的域名隐私泄露风险评估是隐私保护的前提，通过评估的结果实现不同程度隐私泄露风险域名的访问控制，从而达到用户隐私保护的目的。

本文以 Android 平台为例，实现了基于虚拟专用网络服务（VPN Service, virtual private network service）框架的 App 流量抓取 HostRisk 客户端和后台服务器，通过实验验证了该方法的有效性。本文的主要贡献如下。

1) 提出一种基于 TF-IDF 模型和层次聚类方法的隐私泄露评估的方案，通过域名的行为特征等考虑其危害程度。

2) 基于 Android 4.0 版本及其更高版本提供的 VPN Service 框架，实现了用户移动智能终端 App 流量特征提取客户端。

3) HostRisk 方案细粒度地考量 App 发送每个数据分组的行为特征，评估信息接收方域名的危害程度。

2 相关工作

移动网络中的用户隐私泄露问题随着移动设备的激增而日益突出，特别是针对 Android 操作系统。已有研究[12-14]揭示了基于 Android 安全模型开发应用程序的漏洞和威胁，越来越多的学者研究 Android 系统及其衍生产品中的用户隐私信息泄露问题。其中，第三方库检测[15-23]和恶意应用程序检测[24-32]是 2 个研究热点。

2.1 第三方库检测

早期的研究热点主要关注第三方检测中广告库的识别问题。2012 年，Grace 等[6]通过收集已知的广告库设置白名单，匹配 App 源代码中的分组名识别广告库。白名单的方式在小范围内简单有效，但是针对超大规模数据集，工作量大幅增加，并且无法检测出匿名的广告库。2013 年，Book 等[15]分析了 114 000 款 App 广告库的行为特征，揭示了广告库大量地并行存在于多个 App 中，并进一步统计分析了广告库中的 API 调用关系、权限申请等行为特征。

近期，学者们使用机器学习、统计模型、特征识别等手段优化第三方库检测问题。2014 年，Naraynan 等[16]提出了基于语义分析的机器学习检测方案，根据分组名的语义及依赖关系使用支持向量机（SVM, support vector machine）线性分类器进行检测；同年，Sun 等[17]首次提出针对 Android 模型的 Native 层进行第三方调用的隐私保护方案，通过对 Native 层的 API 调用管理和权限申请管理避免第三方从 Native 层获取隐私信息。2016 年，Backes 等[18]提出了一种可靠的大规模第三方库检测方式，

将.dex 压缩文件中 Class 分解为 hash 树，树的叶子节点是一个类的 hash 值。在大规模应用的比对中，匹配到相同的 hash 值即为同一个服务提供商的软件开发包（SDK, software development kit）。在此基础上，根据匹配的相似度进一步分析了同一库的不同版本更新速度，从而达到第三方库检测的目的。Crussel 等[19]和 Wang 等[20]提出 AnDarwin 和 WuKong 方案，通过开发库代码聚类技术检测克隆 App，这类方法依赖于大量应用程序普遍使用开发库进行开发的假设，并且开发人员不会修改开发库，因此这种假设几乎不成立，因为在 App 开发期间删除不必要的代码必然会修改开发库。Li 等[8]和 Ma 等[9]基于 WuKong 的聚类方法提出 LibD 和 LibRadar 的方案，该类方案将混淆的分组名称考虑在内，LibRadar 不比较开发库二进制之间的差异，而是通过特征散列对候选库进行分类，根据分组的目录结构识别开发库。LibRadar 要求构造分组层次结构子树，而这个假设显然是不切实际的，因为一个开发库可以封装在不同的分组中。王浩宇等[21]提出基于多级聚类的方案，对大量应用代码分组粒度提取 API 特征进行聚类，通过权限申请、API 调用等特征进行机器学习分类，最终识别第三方库，其准确率达到了 84.28%。

第三方库采集用户隐私信息是造成隐私泄露的主要原因之一，现有工作几乎都是围绕第三方库的静态代码检测方案，静态检测的方案具有准确率高的优点，但针对第三方库名称动态变化等场景不具备灵活性而且需要选取稳健性较高的特征信息。

2.2　恶意应用程序检测

恶意应用程序是用户在无意识或无法辨别恶意软件的情况下安装的，安装后可能会中断设备的功能或者在用户未能察觉的情况下执行恶意代码。针对恶意应用程序的检测大体有 2 种解决思路：静态分析和动态检测。静态分析是在不运行 App 的情况下对 App 的代码特征进行分析和判断；动态检测方案通过模拟恶意应用程序运行，捕获恶意应用程序的行为特征，并对其进行识别和检测。

静态分析恶意软件的方法受到静态程序分析概念的启发。许多研究将应用程序反编译后进行代码的静态解析[26-28]。Enck 等[24]提出 Kirin 模型，通过分析应用程序申请使用系统权限的情况，研究恶意应用程序的行为特征。Seo 等[25]首先分析了已知恶意软件的异常行为相关的风险 API 和关键字，从解压缩的应用程序文件中提取权限信息，并在应用程序中搜索这些有风险的 API 和关键字的存在，从而判断恶意应用程序。

动态监测方案中，Tenenboim-Chekina 等[26]关注了应用自更新过程中访问恶意域名的问题，他们在特定时间间隔内收集了 App 的特征数据并且使用聚合方法进行计算，基本思想就是找到特征之间的关系，分析恶意流量特征与正常流量特征之间的偏差，从而找出恶意流量。Zhou 等[27]提出了一种名为 DroidRanger 的动态检测器，为了检测已知的恶意软件，首先进行了基于权限的过滤方法，查找应用程序中的危险权限；针对未知恶意软件检测，使用了基于启发式的过滤方法，查找 App 中的可疑行为。李梦玉等[28]利用时间序列的分析方法建立了一种基于 URL 的恶意访问检测模型，首先以一段时间内用户访问某域名的 URL 日志为分析单位，衍生出识别恶意访问的特征，利用时间序列的时域和频域的处理方法将其访问日志向量化，最后通过聚类的方法识别恶意 URL。李佳等[29]基于原始数据与经验特征工程相结合的思想提出了一种混合结构深度神经网络，标注了一套由 45 万余条恶意流量和 2 000 万余条非恶意流量组成的数据集，此数据集上的准确率达到了 98.1%~99.99%。静态分析和动态检测是目前最有效的 2 种检测方案。静态检测方案正确率高但是可操作性弱，不能灵活地应对动态场景。动态检测方案有很高的正确性和可操作性。本文通过 App 的网络流量特征，提出基于 TF-IDF 模型和层次聚类方法的动态检测方案。

2.3　相关工作总结和对比

第三方库检测和恶意应用程序检测是移动端用户隐私泄露检测及评估的研究热点，静态分析和动态检测成为了主要的技术手段。表 1 对比了有代表性的相关工作，特征统计和关键字匹配是最常用的方案，也相对简单，但是存在覆盖率低和准确率低的问题。机器学习的兴起使问题有了新的解决途径，聚类是最常见的机器学习方案，Ma 等[9]和 Tenenboim-Chekina 等[26]方案的粗粒度聚类造成准确率不高的问题。李梦玉等[28]的神经网络方案将问题聚焦在 URL 日志上，并没有考虑到更细粒度的流量特征。

3　预备知识

3.1　免费 App 服务提供商盈利模式

诸如广告投放、用户数据收集等活动的非用户

表 1 现有相关工作对比

代表工作	检测方法	特征	检测类别	粒度	覆盖率	准确率
Grace 等[6]	特征统计	行为	广告库	App	低	低
Ma 等[9]	聚类	分组依赖关系	第三方库	开发分组	低	中
Seo 等[25]	特征统计	关键字	恶意 App	开发分组	低	低
Teneboim-Chekina 等[26]	聚类	流量特征	恶意 App	数据分组	中	高
李梦玉等[28]	神经网络	URL 日志	恶意 App	数据分组	中	高

主动支付而产生的利润，往往是由 App 服务提供商允许广告商等在其 App 客户端中嵌入第三方库所产生的。本文将这些嵌入 App 客户端中的第三方库称为非主业务的第三方库（域名），这些非主业务第三方库将继承 App 客户端的所有权限，直接在用户移动智能终端获取用户的信息，App 服务提供商通过这种方式间接获取利润。

用户在同意 App 相关隐私政策后将 App 服务提供商视为可信服务商，App 服务提供商获取用户数据后有义务保护用户的特定数据，不能用以谋取商业利润。因此更多的 App 选择允许非主业务第三方库嵌入 App 客户端，进而间接从第三方获利。市场上大量免费应用依赖此类盈利模型谋取利润，因此“免费”的 App 实际上并不“免费”。用户在使用 App 时并不能及时检测出个人信息是否被采集，更无法得知个人信息的去向。利用用户的隐私信息换取 App 的服务对用户而言往往不能接受。

3.2 TF-IDF 模型

TF-IDF 是一种统计方法，广泛应用在信息检索、自然语言处理等领域中，是关键字抽取、自动标签生成等问题的常用解决方案。该方法用以评估一个字词对于一个文件集或语料库中一份文件的重要程度。字词的重要性与它在文件中出现的次数成正比，但同时与它在语料库中的逆向文档频率成反比，该模型主要包含 2 个因素，具体介绍如下。

1) 词 W 在文档 D 中的词频（TF, term frequency），表示词 W 在文档 D 中出现次数 $\mathrm{Count}(W,D)$ 和文档 D 中总词数 $\mathrm{Size}(D)$ 的比值，即

$$\mathrm{TF}(W,D)=\frac{\mathrm{Count}(W,D)}{\mathrm{Size}(D)} \tag{1}$$

2) 词 W 在整个文档集合中的逆向文档频率（IDF, inverse document frequenc），即文档总数 N 与词 W 所出现文件数 $\mathrm{Docs}(W,D)$ 比值的对数。

$$\mathrm{IDF}=\log\left(\frac{N}{\mathrm{Docs}(W,D)}\right) \tag{2}$$

TF-IDF 模型根据式(1)的 TF 值和式(2)的 IDF 值为每一个文档 D 和由 k 个关键词 $W_1,\cdots,W_k$ 组成的查询串 q 计算一个权值，用于表示查询串 q 与文档 d 的匹配度。

$$\text{TF-IDF}(q,D)=\sum_{i=1}^{k}(\mathrm{TF}(W_i,D)\mathrm{IDF}(W_i))$$

3.3 App 内主业务域名

一个 App 承载了不同的功能和业务，对应不同的域名为这些功能和业务做数据支撑，如某新闻类 App，该 App 会与多个域名进行数据交互，这些域名包括提供新闻内容的服务器、提供位置信息等的工具类服务器、提供广告的服务器和用于分析用户数据提升服务质量的服务器。其中提供新闻内容的域名服务器承担着该 App 的主要业务功能，App 会频繁大量地与这类域名服务器进行数据交互。而类似广告域名服务器在 App 内不承担主要的业务功能，且几乎这种非主业务的第三方域名服务器不是注册在该 App 服务提供商公司旗下的，因此 App 与该类域名服务器进行数据交互的频率低，数据通信量小。

4 基于流量特征的隐私泄露评估方案

本节介绍所提出的 HostRisk 系统架构，以及相应的客户端和服务器。

4.1 HostRisk 系统架构

HostRisk 系统架构如图 1 所示，其中用户在其智能终端安装 HostRisk 客户端及 TCP 代理，基于 VPN Service 框架的 HostRisk 客户端用于整合所有 App 流量以及修改数据分组的 IP 分组头，TCP 代理用于转发所有的数据分组以及统计流量特征，并发往 HostRisk 服务器。图 1 中，① 代表 App 数据分组首先发往 HostRisk 客户端中的虚拟网卡；② 代表 HostRisk 客户端通过读取虚拟网卡获取 App 数据分组并在修改 IP 分组头地址后将数据分组发送至 TCP 代理的数据转发模块，由数据转发模块将 App

的数据分组发送至 App 服务器；③ 代表 TCP 代理中的特征收集模块统计 App 数据分组的特征数据并汇总发送至 HostRisk 服务器。HostRisk 服务器收集特征数据，首先对数据进行分类和统计，进而使用 TF-IDF 模型和层次聚类方法计算 App 隐私泄露风险。

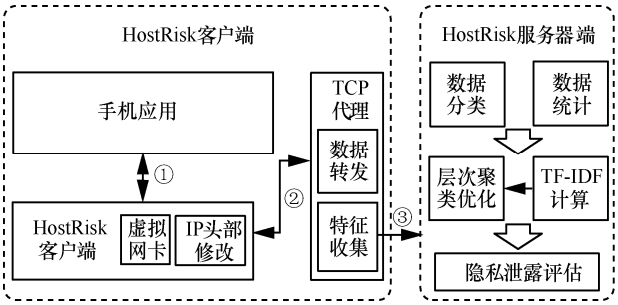


图 1　HostRisk 系统架构

4.2　HostRisk 客户端

基于 Android 4.0 版本以后提供的 VPN Service 框架开发的 HostRisk 客户端收集移动设备中 App 的流量特征。HostRisk 客户端会创建一个虚拟网络接口（虚拟网卡），并返回一个文件描述给 HostRisk 客户端。通过配置地址和路由规则将宿主设备的所有流量转发至虚拟网络接口。HostRisk 客户端通过读取文件描述获取所有 App 发送至 App 对应远端服务器的数据分组，修改数据分组头的 IP 地址和端口号（如图 2 中过程①），将目的地址和端口号修改为 TCP 代理的地址和端口号，将原始地址修改为远端服务器地址，并重新计算数据分组校验和，将数据转发至 TCP 代理，再由 TCP 代理将数据分组发送至 App 对应的远端服务器（图 2 不展示 TCP 代理转发数据分组的流程）。

App 远端服务器的回复数据分组首先发送至 TCP 代理，TCP 代理将数据分组转发至对应的 App，数据分组首先发送至 HostRisk 客户端创建的虚拟接口中，HostRisk 修改数据分组 IP 分组头（如图 2 中过程②），将源地址和端口号修改为远端服务器地址和端口号，目的地址修改为宿主设备地址，然后将数据分组转发至 App。App 发送出的数据分组的目的地址为远端服务器地址，接收到的数据分组的源地址为远端服务器地址，因此 HostRisk 客户端对 App 和远端服务器不可见。图 2 中 A 表示宿主移动设备，B 表示 TCP 代理，C 表示 App 对应远端服务器，D 表示目的地址，S 表示源地址。

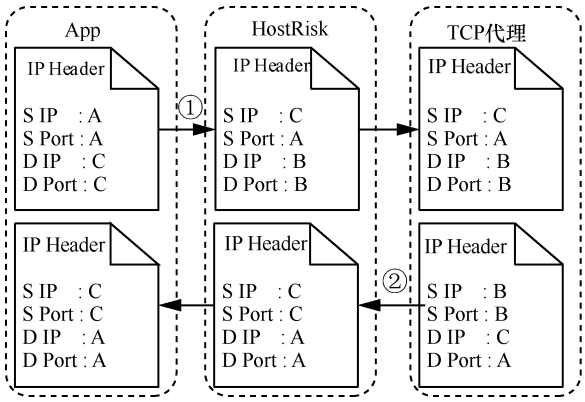


图 2　HostRisk 客户端修改数据 IP 分组头过程

HostRisk 客户端对系统要求的最低版本为 Android 4.0，所需申请的权限有 READ_E-XTERNAL_STORAGE、WRITE_EXTERNAL_ST-ORAGE、ACCESS_WIFI_STAGE、INTERNET 等权限，用户设备的 VPN 权限在 App 运行时申请。HostRisk 客户端通过对 App 收发数据分组流量的监控，将移动设备 App 数据分组特征信息发送至 HostRisk 服务器，并依次计算 App 隐私泄露风险。

4.3　HostRisk 服务器端

与 App（官方渠道下载安装的非恶意 App）进行数据交互的域名不仅是该 App 服务提供商旗下承担主要业务的域名服务器，还存在广告商域名服务器、第三方数据统计域名服务器和第三方工具类域名服务器等。对于非恶意的 App 来说，与之进行数据分组交互个数越多的域名服务器越有可能承担该 App 的主要业务，图 3 是 5.2 节所述数据集中某新闻类 App 与所有域名服务器收发数据分组个数的柱状图，其中 pstatp.com、ixigua.com 等均注册在该 App 所属母公司旗下，该图中的 doubleclick.com、google-statics.com、umeng.com 等是著名的推送以及信息采集、统计的服务商。其中，App 与非主业务第三方域名服务器交互数据分组个数对于主业务域名服务器交互数据分组个数几乎可以忽略不计。

4.3.1　基于 TF-IDF 的域名业务相关性计算

Book 等[15]研究发现广告商、信息推送商、信息采集商等非主业务第三方域名广泛地存在于多个 App 之间，且通信数据量较 App 主业务域名服务器的通信数据量小。域名服务器的业务相关性与 App 和该域名交互数据分组个数成正比，同时与该域名服务器出现在所有 App 中的频率成反比。本文基于 TF-IDF 模型提出了域名与 App 业务相关性分值的计算方法，具体如下。

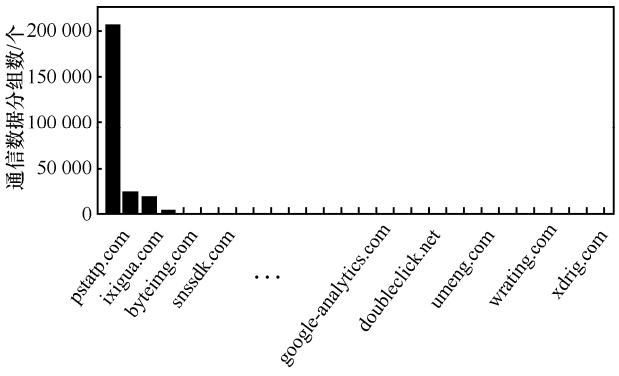


图 3 某新闻类 App 与各域名交互数据分组个数统计

$$\mathrm{HA}(h)=\frac{\mathrm{HCount}(h)}{\mathrm{HCount}(H)}\log\left(\frac{\mathrm{ACount}(\mathrm{App})}{\mathrm{ACount}(\mathrm{APPs}(h))}\right)$$

其中，HA(h) 表示计算域名的业务相关性分值，H 是该 App 中所有域名的集合，HCount() 函数用于统计该 App 与域名交互的数据分组个数，App 表示 HostRisk 系统收集的所有 App 集合，APPs(h) 函数用于统计所有包含域名 h 的 App 并返回集合，ACount() 用于统计 App 集合中 App 的个数。

基于 TF-IDF 模型计算 App 内所有域名的业务相关性分值，通过该值将 App 内的域名进行排序，排序结果中域名的分布如图 4 所示。App 内承担主要业务的域名几乎分布在 App 主业务域名区中，非主业务的第三方域名如广告商、推送商等分布在第三方域名区中，混杂区中混合着上述 2 种域名。基于 TF-IDF 模型的排序结果精度随着混杂区的增大而降低，进而本文通过平均连接的凝聚型层次聚类方法进行优化调整。

图 4 基于 TF-IDF 计算后的排名结果

4.3.2 基于层次聚类的优化算法

基于 TF-IDF 模型的排序方法能够大致地将 App 内域名按业务相关性进行排名，但存在的问题是混杂区中同时混合着 App 内承担主业务的域名和非主业务的域名。本文使用平均连接的凝聚型层次聚类的优化模型来减小混杂区以提高方案的准确性。凝聚型层次聚类的优点是不需要指定聚类个数，当所有元素聚到同一个类中或者最小相似度达到某一阈值时，聚类结束。

TCP 通道是 App 与域名服务器传输数据的载体，TCP 通道中数据传输的流量特征反映出域名采集数据的流量特征，TCP 通道中传输的数据分组头部会记录域名服务器的 IP 地址端口等信息，数据分组的分组体中会记录域名服务器的域名地址，域名服务器与 TCP 通道呈现一对多的关系。

本节将研究对象转换到 App 与域名所建立的每一个 TCP 通道上，在更细粒度维度上研究它们的行为特征。混杂区域中既包括主业务域名又包括非主业务域名，而该区域中未能正确计算的主业务域名与分布在 App 主业务域名区中的域名有很高的“相似性”，通过 TCP 通道之间的“相似性”将混杂区中的主业务域名筛选出来。TCP 通道之间的“相似性”定义为：1) 相似的域名后缀；2) 临近的 IP 地址；3) 相似的采集行为（平均数据分组大小、上传下载数据比例等）。本文将从这些特征研究 2 个域名 TCP 通道之间的距离 $\mathrm{Dis}_T(x,y)$。首先根据 TCP 通道之间的“相似性”定义如下相似距离。

1) Host Distance。域名地址是衡量 2 个域名相似性的重要因素。相同的域名后缀代表域名拥有相同所属机构，而相似的域名前缀代表域名具有相似业务。本文将域名地址拆分成 2-gram 集合 $\mathrm{JSet}_{\mathrm{host}}$，并计算 2 个 $\mathrm{JSet}_{\mathrm{host}}$ 的 Jaccard 距离，Jaccard 距离[32]用于比较有限样本集之间的相似性与差异性，具体如下。

$$\mathrm{Jaccard}(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

其中，A 和 B 代表 2 个集合。域名前缀和后缀匹配度越高，其距离越近，计算式定义如式(3)所示，并将其归一化到[0,1]范围内，其中 min()函数返回 2 个数中较小的值。

$$\mathrm{Dis}_H=1-\frac{\mathrm{Jaccard}(\mathrm{JSet}_x,\mathrm{JSet}_y)}{|\min(\mathrm{JSet}_x,\mathrm{JSet}_y)|}\in[0,1] \tag{3}$$

2) IP Distance。互联网编号分配机构（IANA, Internet assigned numbers authority）通常根据机构规模等因素分配 IP 地址，连续的 IP 地址通常属于同一组织机构，通过最长前缀匹配的规则来衡量 IP 地址之间的相似距离。最长前缀匹配算法是获取 2 个 IP 地址二进制串最长的相似位数，计算式

较简单，此处将略过计算式。前缀匹配越长的 IP 距离越近，计算式定义如式(4)所示，并将其归一化至[0,1]范围内，其中 MaxIpPrexMatch()函数返回前缀匹配的最大长度，IPLength 表示 IP 字符串长度。

$$\mathrm{Dis}_I = 1 - \frac{\mathrm{MaxIpPrexMatch}(\mathrm{IP}_x, \mathrm{IP}_y)}{\mathrm{IPLength}} \in [0,1] \tag{4}$$

3) Action Distance。本节将从 App 与域名之间 TCP 通道中数据分组的平均大小（APS, average packet size）和上传下载数据比（ROUD, ratio of upload to download）考察 TCP 通道的相似性，计算式如式(5)所示，并将该结果归一化至[0,2]范围内，其中 Abs()函数返回该数值的绝对值，max()函数返回 2 个数中较大的数值。

$$\mathrm{Dis}_A = \frac{\mathrm{Abs}(\mathrm{APS}_x - \mathrm{APS}_y)}{\max(\mathrm{APS}_x, \mathrm{APS}_y)} + \frac{\mathrm{Abs}(\mathrm{ROUD}_x - \mathrm{ROUD}_y)}{\max(\mathrm{ROUD}_x, \mathrm{ROUD}_y)} \in [0,2] \tag{5}$$

式(3)~式(5)计算出的 Dis_H 、 Dis_I 和 Dis_A 通过加权求和的方法，计算 2 个域名对应 TCP 通道之间的距离，即

$$\mathrm{Dis}_T = w_1\mathrm{Dis}_H + w_2\mathrm{Dis}_I + w_3\mathrm{Dis}_A \in [0,1]$$

其中，Dis_T 为层次聚类方法所采取的分类准则；w_i 是加权值，根据实际场景对不同聚类标准的需求，进行权重的个性化分配，系统默认配置为满足 $\sum w_i = 1$ 条件下的均分权重，并在进行层次聚类前计算 App 中所有 TCP 通道之间的相似距离。App 与每个域名创建的 TCP 通道将会被聚类至一个或多个类别（C）中，根据 TF-IDF 模型计算结果的排名值，计算每个类别（C）的业务相关性得分 CScore 为

$$\mathrm{CScore}(C) = \frac{\sum_{i=0}^{C.\mathrm{size}} \mathrm{Rank}(h_i)}{C.\mathrm{size}} \tag{6}$$

其中， Rank() 函数用于计算域名 h_i 在 TF-IDF 排名表中的排名结果。App 会与某一域名创建多个 TCP 通道，通过 TCP 通道所在类别业务相关性得分，即式(6)计算结果的平均值，并计算域名的业务相关性得分 HScore 为

$$\mathrm{HScore}(h) = \frac{\sum_{i=0}^{h.\mathrm{size}} \mathrm{CScore}(C_i, h \in C_i)}{h.\mathrm{size}} \tag{7}$$

App 中所有域名的业务相关性得分反映域名在 App 内扮演的角色（主业务、广告商等），App 将隐私信息发送至广告商等非主业务第三方域名造成隐私泄露，通过式(7)的计算结果，计算 App 隐私泄露风险值 RiskScore 如下。

$$\mathrm{RiskScore}(\mathrm{App}) = \frac{\sum_{h}^{h \in \mathrm{App}} \mathrm{HScore}(h)}{\mathrm{Count}(h, h \in \mathrm{App})}$$

其中， Count() 函数用于统计 App 内域名的总数。

5　实验及分析

为测试本文提出的隐私泄露评估方案，首先实现了 HostRisk 原型系统，包括一个 Android 客户端和一个后台服务器。Android 客户端安装在用户智能终端，用于收集 App 的流量信息，其中每个数据分组收集的信息包括用户编号、App 名称、域名地址、IP 地址、端口号、通信协议（HTTP/HTTPS）、数据分组发送时刻、TCP 通道创建时刻、发送方向（上传或下载）、数据分组大小。服务器负责接收客户端发来的数据，整合多个用户多个 App 收发数据分组特征数据，发送至 HostRisk 后台服务器计算其隐私泄露威胁程度。

5.1　数据集

实验征集了 25 名志愿者，进行了 14 天的数据收集，收集了 1 082 060 条数据，共涉及 112 款 App，收集到 4 056 个域名。其中，有 349 755 个 HTTP 数据分组和 732 305 个 HTTPS 数据分组，比例为 32.32% 和 67.68%，上传总量为 55 352 806 B，下载总量为 2 251 886 263 B，总传输量为 2 307 239 069 B。

5.2　业务属性计算方法评估

App 中的所有域名参与计算，通过 TF-IDF 模型和层次聚类方法计算所有域名的业务相关性，进而计算 App 的隐私泄露威胁值。其中 App 主业务域名的相关性分值低（分值越低代表相关性越高），而非主业务的第三方域名相关性分值高，判断其正确性。

5.2.1　域名相关性计算排序表分析

实验选取了某公司旗下著名的新闻类 App，该 App 是一款基于数据挖掘的推荐引擎产品。截至

2019 年 6 月 10 日，“七麦数据网”发布该 App 共有 10 176 786 466 次下载量，近 30 天的日均下载量为 6 442 101 次，研究该 App 的隐私泄露程度具有一定的代表性。为了使聚类效果最好，聚类数目最佳，将阈值设置为 0.8 效果最佳。通过多次实验观察，发现域名地址对于实际聚类的影响大，因此将距离计算中的权重按表 2 进行分配，其中 w_1 表示域名地址之间的距离，是衡量 2 个域名 TCP 通道之间相似性的重要指标；w_2 表示域名 IP 之间的距离；w_3 表示域名 TCP 通道的行为特征距离。

表 2　　相似距离权重分配

权重	取值
w_1	0.6
w_2	0.2
w_3	0.2

表 3 为某新闻类 App 中所有域名相关性计算的排名和相关性分值结果，其中分值越小则风险越小，分值的取值上限与 App 中域名个数成正比。pstatp.com、snssdk.com、bytecdn.cn、byteimg.com 均注册在该新闻类 App 所属母公司旗下，用于提供新闻信息提供、用户日志记录、内容缓存等服务；ixigua.com 注册在隶属于该母公司旗下的子公司，用于提供该 App 中视频服务等。通过相似性计算后这 5 个域名的相似性值位列前五，隐私风险得分分别为 1.0、2.0、5.0、4.0、12.0，其代表该 App 的主要业务。google-analytics.com、xdrig.com、amap.com、wrating.com、doubleclick.net 是著名的数据统计和第三方工具类域名，根据计算结果排序为该新闻类 App 中相关性分值倒数五名的域名，表 4 是这些域名的数据特征。xdrig.com 是某数据科技公司旗下著名的数据采集分析平台；amap.com 是某地图类科技公司旗下的域名平台；wrating.com 注册在某数据有限公司旗下，用于收集分析数据的域名；google-analytics.com 和 doubleclick.net 隶属于某著名公司旗下，用于收集数据的第三方平台。按排序从高到低的得分依次为 48.92、50.0、56.23、58.0、69.5，相比之下这些域名被视作第三方域名，隐私泄露风险高。

表 3　某新闻类 App 域名相关性计算排名结果

域名地址	注册机构名称	分值
pstatp.com	App 所属公司	1.0
ixigua.com	App 所属公司	2.0
snssdk.com	App 所属公司	5.0
byteimg.com	App 所属公司	4.0
bytecdn.cn	App 所属公司	12.0
⋮	⋮	⋮
wrating.com	第三方数据科技公司	48.92
doubleclick.net	第三方数据科技公司	50.0
google-analytics.com	第三方数据科技公司	56.23
amap.com	第三方数据科技公司	58.0
xdrig.com	第三方数据科技公司	69.5

表 4 是某新闻类 App 中原始数据中隐私泄露风险较高域名流量的统计数据，这些域名大多是广告域名或用于统计用户信息的域名，由统计数据易知该 App 与这些域名之间的通信数据分组个数少且通信数据量低，非主业务第三方域名大量使用 HTTPS 进行数据通信。除 amap.com 外，其他用于收集用户数据的域名上传数据量均大于下载数据量，amap.com 属于第三方工具类域名上传数据量和下载数据量持平。通过 TF-IDF 模型和层次聚类计

表 4　　某新闻类 App 中倒数 5 个业务不相关域名通信数据特征

域名地址	IP 地址	通信协议				通信量					平均数据分组大小/B
		HTTP 数据分组		HTTPS 数据分组		上传量		下载量		上传下载比	
		个数	占比	个数	占比	总量/B	占比	总量/B	占比		
xdrig.com	52.80.186.222	0	0	1	5.76×10^{-6}	32	5.73×10^{-8}	0	0	32	32
amap.com	106.11.186.5	18	1.7×10^{-4}	12	6.91×10^{-5}	13 320	2.46×10^{-5}	14 219	0.002 7	0.94	917.97
google-analytics.com	203.208.40.39	0	0	105	6.05×10^{-4}	88 434	1.64×10^{-4}	5 747	0.001 1	15.38	896.96
doubleclick.net	203.208.41.77	0	0	17	9.8×10^{-5}	6 735	1.25×10^{-5}	1 251	2.4×10^{-4}	5.38	469.76
wrating.com	106.11.12.3	0	0	11	6.34×10^{-5}	9 803	1.81×10^{-5}	674	1.29×10^{-4}	14.54	952.45
综合	—	70 182	—	173 475	—	540 411 034	—	5 219 493	—	—	2 239.33

算后，这 5 个非主业务第三方域名排在倒数五名，该 App 内域名隐私泄露危害风险根据排序表得出，其中 xdrig.com 等域名泄露用户隐私危害风险大，pstatp.com 等域名泄露用户隐私危害风险小，反映在表 3 中的隐私泄露风险分值计算中，隐私泄露风险高的域名分值低。

5.2.2 算法性能评估

数据量大小和层次聚类阈值的选取是算法计算时间复杂度的主要影响因素，本节针对数据量的大小和层次聚类的阈值对算法时间复杂度进行性能评估实验，实验结果如表 5 所示。如 5.1 节所述，数据集中进行随机删除构造 1 万、10 万、50 万和 100 万的不同量级数据集。实验选取在一台搭载 Windows 7 操作系统、6 GB 内存容量、CPU 主频为 3.5 GHz 的 PC 机上进行。由表 5 可知，数据量的大小成为影响算法计算时间复杂度的主要因素，由于平均连接的凝聚型层次聚类算法会迭代计算 App 内所有域名创建 TCP 通道之间的相似性距离，并计算多个类之间平均相似聚类最小值，迭代计算最小值的过程消耗大量时间，当数据量多达 100 万时，计算时间在 100 s 以内。

表 5　　算法性能评估

阈值	1 万/s	10 万/s	50 万/s	100 万/s
0.10	0.005	0.648	7.940	63.268
0.25	0.004	0.645	10.048	80.436
0.50	0.005	0.664	10.806	87.788
0.75	0.004	0.756	11.727	90.650
1.00	0.004	0.672	11.052	89.660

6 结束语

针对 Android 平台上 App 内非主业务第三方域名采集用户隐私信息造成隐私泄露问题，本文提出了一种移动设备中基于流量特征的隐私泄露评估方案。基于 TF-IDF 模型和平均连接的凝聚型层次聚类方法计算所有域名与 App 的业务相关性得分，相关性得分越低的域名与 App 的主业务越相关，而相关性得分越高的域名造成隐私泄露的风险越大，最终通过加权平均的方法计算 App 的隐私泄露风险。其中域名的相关性得分可判定域名在 App 内扮演的角色（主业务、广告商等），进而设置隐私保护方案以保证用户的隐私信息不被泄露，在保证 App 服务质量的同时降低用户隐私信息泄露带来的威胁。本文实现了隐私泄露评估 HostRisk 的客户端和后台服务器，并在一组真实的实验数据上进行了测试，实验结果进一步说明了该方法的有效性和效率。

参考文献：

[1] LI F H, LI H, NIU B, et al. Privacy computing: concept, computing framework, and future development trends[J]. Elsevier Engineering, 2019, 5(6): 1179-1192.

[2] REN J, RAO A, LINDORFER M, et al. Recon: revealing and controlling PII leaks in mobile network traffic[C]//The 14th Annual International Conference on Mobile Systems, Applications, and Services. ACM, 2016: 361-374.

[3] WANG H, GUO Y. Understanding third-party libraries in mobile App analysis[C]//2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C). IEEE, 2017: 515-516.

[4] BOOK T, WALLACH D S. A case of collusion: a study of the interface between ad libraries and their Apps[C]//The Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices. ACM, 2013: 79-86.

[5] STEVENS R, GIBLER C, CRUSSELL J, et al. Investigating user privacy in android ad libraries[C]//Workshop on Mobile Security Technologies (MoST). Citeseer, 2012: 10.

[6] GRACE M C, ZHOU W, JIANG X, et al. Unsafe exposure analysis of mobile in-App advertisements[C]//The Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks. ACM, 2012: 101-112.

[7] LIN J, AMINI S, HONG J I, et al. Expectation and purpose: understanding users' mental models of mobile App privacy through crowdsourcing[C]//The 2012 ACM Conference on Ubiquitous Computing. ACM, 2012: 501-510.

[8] LI M, WANG W, WANG P, et al. LibD: scalable and precise third-party library detection in android markets[C]//2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). ACM, 2017: 335-346.

[9] MA Z, WANG H, GUO Y, et al. LibRadar: fast and accurate detection of third-party libraries in Android Apps[C]//The 38th International Conference on Software Engineering Companion. 2016: 653-656.

[10] KUZUNO H, TONAMI S. Signature generation for sensitive information leakage in android applications[C]//2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW). IEEE, 2013: 112-119.

[11] LI J, ZHAI L, ZHANG X, et al. Research of android malware detection based on network traffic monitoring[C]//2014 9th IEEE Conference on Industrial Electronics and Applications. IEEE, 2014: 1739-1744.

[12] HE Y, YANG X, HU B, et al. Dynamic privacy leakage analysis of android third-party libraries[J]. Journal of Information Security and Applications, 2019, 46: 259-270.

[13] FANG Z, HAN W, LI Y. Permission based Android security: issues and countermeasures[J]. Computers & Security, 2014, 43: 205-218.

[14] ENCH W, OCTEAU D, MCDANIEL P D, et al. A study of Android application security[C]//USENIX Security Symposium. 2011: 2.

[15] BOOK T, PRIDGEN A, WALLACH D S. Longitudinal analysis of android ad library permissions[J]. arXiv Preprint, arXiv:1303.0857, 2013.

[16] NARAYNAN A, CHEN L, CHAN C K. Addetect: automated detection of Android ad libraries using semantic analysis[C]//2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). IEEE, 2014: 1-6.

[17] SUN M, TAN G. Nativeguard: protecting android applications from third-party native libraries[C]//The 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks. ACM, 2014: 165-176.

[18] BACKES M, BUGIEL S, DERR E. Reliable third-party library detection in android and its security applications[C]//The 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 356-367.

[19] CRUSSEL J, GIBLER C, CHEN H. Andarwin: scalable detection of semantically similar android applications[C]//European Symposium on Research in Computer Security. 2013: 182-199.

[20] WANG H, GUO Y, MA Z, et al. Wukong: a scalable and accurate two-phase approach to android app clone detection[C]//The 2015 International Symposium on Software Testing and Analysis. 2015: 71-82.

[21] 王浩宇, 郭耀, 马子昂, 等. 大规模移动应用第三方库自动检测和分类方法[J]. 软件学报, 2017, 6: 1373-1388.
WANG H Y, GUO Y, MA Z A, et al. Automated detection and classification of third-party libraries in large scale Android Apps[J]. Journal of Software, 2017, 6: 1373-1388.

[22] LIU B, LIU B, JIN H, et al. Efficient privilege de-escalation for ad-libraries in mobile Apps[C]//The 13th Annual International Conference on Mobile Systems, Applications, and Services. 2015: 89-103.

[23] TANG Z, XUE M, MENG G, et al. Securing Android applications via edge assistant third-party library detection[J]. Computers & Security, 2019, 80: 257-272.

[24] ENCK W, ONGTANG M, MCDANIEL P. On lightweight mobile phone application certification[C]//The 16th ACM conference on Computer and Communications Security. ACM, 2009: 235-245.

[25] SEO S H, GUPTA A, SALLAM A M, et al. Detecting mobile malware threats to homeland security through static analysis[J]. Journal of Network and Computer Applications, 2014, 38: 43-53.

[26] TENENBOIM-CHEKINA L, BARAD O, SHABTAI A, et al. Detecting application update attack on mobile devices through network features[C]//2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2013: 91-92.

[27] ZHOU Y, WANG Z, ZHOU W, et al. Hey, you, get off of my market: detecting malicious apps in official and alternative android markets[C]//19th Annual Network & Distributed System Security Symposium. 2012: 50-52.

[28] 李梦玉, 马严, 黄小红, 等. 基于 URL 的恶意访问检测方法[J]. 通信学报, 2018, 39(Z1): 92-98.
LI M Y, MA Y, HUANG X H, et al. Malicious access detection method based on URL[J]. Journal on Communications, 2018, 39(Z1): 92-98.

[29] 李佳, 云晓春, 李书豪, 等. 基于混合结构深度神经网络的 HTTP 恶意流量检测方法[J]. 通信学报, 2019, 40(1):28-37.
LI J, YUN X C, LI S H, et al. HTTP malicious traffic detection method based on hybrid structure deep neural network[J]. Journal on Communications, 2019, 40(1):28-37.

[30] GRACE M, ZHOU Y, ZHANG Q, et al. Riskranker: scalable and accurate zero-day Android malware detection[C]//The 10th International Conference on Mobile Systems, Applications, and Services. 2012: 281-294.

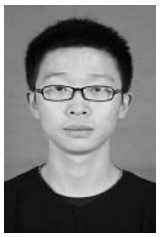
[31] KUMAR R, ZHANG X, WANG W, et al. A multimodal malware detection technique for Android IoT devices using various features[J]. IEEE Access, 2019, 7: 64411-64430.

[32] ALSWAINA F, ELLEITHY K. Android malware permission-based multi-class classification using extremely randomized trees[J]. IEEE Access, 2018, 6: 76217-76227.

[33] LEVANDOWSKY M, WINTER D. Distance between sets[J]. Nature, 1971, 234(5323): 34.

[作者简介]

王竹（1972− ），女，山西太原人，博士，中国科学院高级工程师，主要研究方向为信息安全、人工智能。



贺坤（1995− ），男，安徽安庆人，中国科学院硕士生，主要研究方向为信息保护、隐私计算。

王新宇（1989− ），男，甘肃平凉人，中国科学院博士生，主要研究方向为信息保护、隐私计算。

牛犇（1984− ），男，陕西西安人，博士，中国科学院副研究员，主要研究方向为网络安全、隐私计算。

李凤华（1966− ），男，湖北浠水人，博士，中国科学院研究员、博士生导师，主要研究方向为网络与系统安全、信息保护、隐私计算。