对逆向出手机应用程序收集用户敏感信息潜在目的的技术学习和分析

by 骆信智 08163337

目录

> 以下报告为对 ACM 期刊《Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)》2018年12月刊登的《"Why Are They Collecting My Data?": Inferring the Purposes of Network Traffic in Mobile Apps》一文的概述、总结与思考，包括对论文总体结构、数据隐私分析处理、机器学习在信息安全方面应用具体手段、安卓设备流量逆向、目的分类方法论等实践的学习理解。
>
> 项目源码地址：https://github.com/CMUChimpsLab/MobiPurpose

摘要

许多智能手机应用程序收集潜在的敏感个人数据，并将其发送到云服务器。然而，大多数移动用户对他们的数据被收集的原因知之甚少。

遵循安全和隐私中隐私保护的理念，文章作者提出了MobiPurpose，这是一种新的技术，它可以接收Android应用程序发出的网络请求，然后对数据收集的目的进行分类，作为向非专家（手机用户）解释数据公开上下文的一个步骤。

目的推断通过利用两个观察结果:

1)开发人员命名约定(例如URL路径)通常提供关于数据收集目的的提示

2)外部知识，如app元数据和域名信息，是有意义的线索，可以用来推断不同流量请求的行为。

MobiPurpose将每个流量请求体解析为键值对，并使用监督学习和文本模式引导相结合的方法推断每个键值对的数据类型和数据收集目的。

作者使用由十名人类专家交叉标记的数据集评估了MobiPurpose的有效性。结果表明，MobiPurpose对数据采集目的的预测平均精度为84%（在19个类别中）。

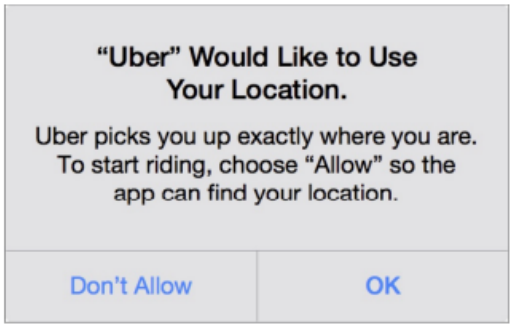
介绍

背景

移动应用程序的一个主要隐私问题是它们可能会访问大量敏感的个人信息。本文作者关注移动隐私的一个维度，即敏感数据离开设备并通过网络发送到远程服务器的节点。许多隐私研究人员已经采用了这个网络视角，研究了隐私敏感的移动数据共享的原始属性，即哪个应用程序在共享数据，共享什么数据，数据将流向何处。

然而，目前很少有研究探究为什么应用程序要求访问敏感数据（也可称为是数据收集的目的）。“为什么”是隐私上下文定义的一个基本组成部分。例如，用户可能更愿意提供他们的位置以使搜索结果更相关，但不太愿意提供位置为了有地区针对性的广告。过去的研究还发现，在不解释原因的情况下展示应用程序正在使用的数据，可能会引发隐私方面的担忧。例如，一些人发现，当最终用户被告知词典应用程序访问了他们的位置时，他们非常担心隐私。然而，当被告知位置数据只是用来搜索附近人们正在搜索的热门词汇时，他们就不那么担心了。另一个例子，人们非常愿意分享他们的位置数据来帮助城市规划公交路线或获得交通信息，但不愿意分享相同的数据用于广告或显示一个人的旅行的踪迹。

这里的一个主要挑战是，目前几乎没有对终端用户理解在智能手机应用程序中使用数据的目的的支持。Android和iOS现在提供了一种功能，可以解释为什么应用程序会通过一个目的字符串或“使用说明”访问敏感用户数据。下图举例说明两个应用程序权限模式对话框示例，其中描述了Camera/Uber(谁)请求位置(什么)来标记照片或定位乘客(为什么)。



但是，这些目的字符串只显示在用户界面层，可以是任意文本。没有方法能验证目的字符串是否准确。

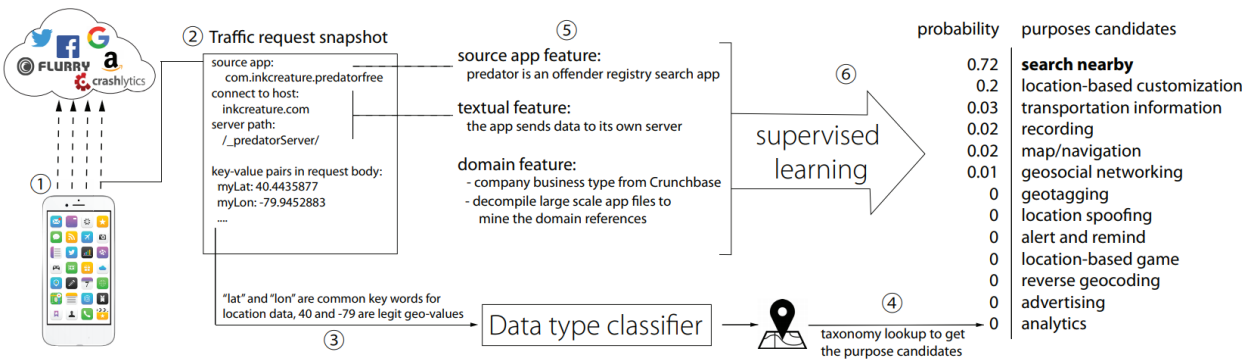
过去有一些关于如何推断目的的工作，例如基于可执行程序的静态文本分析或考虑应用程序使用的库。相反，本文着眼于一种用于推断目的的新方法即基于网络流量，因为研究数据出口为理解从手机流出的特定数据提供了更好的视角。

在本文中提出了一种新的移动通信目的自动推理技术MobiPurpose。MobiPurpose接收智能手机应用程序发出的网络请求，并对每个键值对的用途进行分类，以帮助向非专业人士解释为什么要收集他们的数据。

MobiPurpose的设计初衷是在实验室设备上运行，而不是在终端用户的智能手机上运行。但是作者同时开发了一些工具来自动下载和安装应用程序到智能手机上，并使用Android Monkey与这些应用程序进行交互脚本，并从实验室设备捕获合成网络数据。他们假设network service API从不同的设备收集几乎相同类型的数据。当使用实验室设备截取数据时，获取的信息也可以传递给真实的用户。MobiPurpose是我们实现更大目标的关键一步，我们的目标是提供一个公共的、大规模的数据库(类似于PrivacyGrade)，它不仅可以显示世界各地的不同服务通过智能手机应用程序收集了哪些数据，还可以帮助解释原因。这将使我们以前所未有的方式分析移动应用程序和API服务提供商的隐私实践。

实现方式

MobiPurpose使用VPN方法拦截所有发出的流量请求，并将HTTP(S)请求体解析为KV对。HTTP(S)被认为是智能手机-服务器通信中最常见的协议，最近的研究表明，超过80%的应用程序具有结构化的响应。利用监督机器学习和自然语言处理(NLP)技术的结合，MobiPurpose检查每个KV对的数据公开上下文。下图展示了MobiPurpose推断数据类型为"myLat:40.44;-79.94"表示位置，对应的目的是"搜索附近"。



MobiPurpose的核心是我们的目的分类，在其中我们为每种数据类型显式列举了潜在的数据收集目的。也就是说，我们的目标不是生成描述目的的文本，而是从这个分类中选择适当的目的。对于任何流量请求，我们首先使用引导NLP方法推断涉及哪些数据类型，然后通过分类法查找找到相关的目标候选项。然后，我们使用监督机器学习方法来预测数据收集的目的。

我们的机器学习方法模拟逆向工程专家如何在网络流量中使用各种线索(例如变量名、URL路径等)来推断API细节(例如数据类型、数据收集目的)。我们首先观察到一组专家检查了大量的网络流量请求，并标记了数据类型和用途，然后要求参与者描述他们的推理过程。基于这些观察，我们提出了一组计算特性，利用两个直观事实对这些数据模式进行建模：

- 开发人员的命名约定通常使标识符在KV对和url中都可以自解释。例如，KV对“useradvertising - ID: 901e3310-3a26-487e-83c7-2fa26ac2786c”可能是唯一的ID，因为键名包含“advertising”和“ID”等关键字，并且值是机器生成的UUID。另一个例子是，连接到reports.crashlytics.com的网络请求可能生成崩溃报告，因为域是“崩溃”和“分析”的新词，子域是“报告”。我们还在后文中讨论了潜在的混淆。
- 外部知识，如app描述和领域所有者，可以是推断数据收集目的的有意义的线索。例如，假设我们有一个游戏应用程序向admob.com发送位置数据。由于AdMob是一家移动广告公司，我们可以推断，位置数据可能被用来根据用户的位置定制要显示的广告。

我们使用自动化测试来扩展流量数据收集。我们开发了一个测试工具harness，它可以在Android设备上安装应用程序，然后使用一个随机点击UI元素的界面自动化monkey来探索这些应用程序。在超过50天的时间里，我们使用8台Android设备收集了15k个应用程序的数据，每个应用程序都运行monkey3分钟，拦截了200万个独特的流量请求。然后，我们抽样1059个流量请求，并让10名专家使用我们的What&Why分类法手动标记每个流量请求的行为。由于数据收集的目的可能是主观的和模糊的，我们为每个流量请求收集了3个独立的标签。

我们使用标记数据来评估我们的数据类型推断算法，并训练一个有目的的分类器。在我们的实验中，我们发现MobiPurpose在预测“什么”(在8个独特的数据类型类别中)和预测“为什么”(在19个独特的目标类别中)的平均准确率分别达到95%和84%。

如前所述，我们的方法是为了复现和逆向当开发人员已经做了什么后的api，所以我们的方法不为故意混淆视听的变量名或服务器名称负责，或者数据到远程服务器之后的其他隐藏的其他使用目的。不过，我们相信，MobiPurpose可以成为提高智能手机应用程序生态系统透明度的一个有效起点，方法是规划出哪些数据正在被共享，以及为什么要收集这些数据。

论文贡献

我们的贡献在于目的推理的技术实现。MobiPurpose是解决移动网络目标自动推理的第一项工作。为此，我们首先将目的描述生成任务转化为分类任务，使机器推理成为可能。然后，我们提出了实用的计算方法来模拟逆向工程专家如何在网络流量中使用各种线索(例如变量名、URL路径等)来推断数据收集的目的。我们的具体贡献如下：

- 我们介绍了第一个能够自动对移动网络流量中敏感数据的数据采集目的进行分类的系统的设计与实现。为此，我们开发了一个可扩展的目的分类，并将目的描述生成任务转换为分类任务。
- 为了便于推断，我们提出了一组实用的计算特性和模式。我们研究了不同类型的功能的有效性，表明基于文本的功能和域功能提供了高收益，而源应用程序功能提供了边际改进。
- 我们使用一个包含1059个标记的网络数据实例的人工标记数据集来评估我们的框架815个不同的应用，涉及636个不同的域名。我们发现MobiPurpose对于数据类型推理的平均准确率为95%，对于数据目的的推理的平均准确率为84%。

构建目的分类

本节描述我们的目的分类的设计和开发，它将目的描述生成任务转换为分类任务。这个分类的目标是提供一组全面的目的(包括绝大多数的敏感数据的用例),有一个有意义的粒度(给定目的不应该太窄仅描述应用很少,也不太宽),和可以理解的(开发人员和终端用户可以理解每个目的以最小的解释)。

现有的目的分类

我们首先对现有的用途分类法进行了广泛的调查。我们发现，在过去的工作中，两种不同类型的目的被混合使用。其中一些目的是明确地描述特定的上下文(例如，附近的搜索、地图导航)，而其他的则是更一般的(例如，合法的、主要的)。

我们的目标不是争论这种对隐私敏感的数据披露是否合法。相反，我们希望与非专家沟通数据收集上下文，并帮助他们理解应该期望的内容。因此，我们选择将重点放在更好地描述功能的目的上，例如广告、SNS等，但排除了"主要"、"内部使用"、"合法"或"核心功能"等目的。我们在后文讨论了这一选择背后的详细理由。

研究方法

通过研究大量智能手机应用程序使用的敏感数据(Android权限)和网络流量行为，我们迭代地开发了我们的分类。

- 允许访问。Android将所有权限分为三个保护级别:正常权限、签名权限、危险权限。请求9个"危险"权限需要用户明确同意。我们根据Android应用程序所需的权限对它们进行了索引，并为每个"危险"权限抽取了100个应用程序，总共得到900个应用程序。然后，我们将应用程序名称、应用程序描述和权限打印出来900张索引卡，将用于以后的卡片排序会话。
- 网络流量。我们在两款智能手机上安装了20个最受欢迎的免费应用程序，每个应用程序我们都会积极使用3分钟，目的是调用这些应用程序的主要功能。我们使用MITM (man-in- middle) VPN应用程序拦截连接到321个惟一域的5504个惟一HTTP(S)流量请求。然后，我们对其中400个流量请求进行了采样，每个请求都联系一个惟一的URL(域+路径)。我们将目标主机名、应用程序名、应用程序描述和数据体打印在400个索引卡上。

我们进行了四次卡片排序会话来创建和改进分类。在每个迭代中，参与者首先独立地创建自己的分类法，然后与其他参与者讨论以达成一致。聚合分类法将是下一个迭代的起点。总的来说，我们在这四个会议中有12名参与者，包括移动开发人员和UX设计师。

在我们早期的分类法迭代中，出现的一个主要问题是如何将目的组织成有意义的类别。

- 我们从一些独立目的类别开始，例如社交/沟通，广告、游戏、多媒体消费、内容、分析、个性化和地图/导航。这种结构对开发人员和用户都是直观的，因为应用程序是通过应用程序商店中的这些类别来组织的。然而，这种目的粒度主要停留在应用程序类别级别，只提供很少的隐私洞察。
- 然后我们切换到另一个层次结构，其中子节点是目的，父节点是应用程序类别。然而，在以后的迭代中，参与者发现这很难使用，因为它能够快速查找目标。任何类别的应用程序都可以出于任何目的发出流量请求。此外，这种结构可以帮助用户了解应用程序的隐私友好程度，但不能帮助用户了解特定网络请求的公开上下文。
- 我们最终的分类灵感来自于app permission模态对话框，我们仍然在最终的分类中使用一个层次结构，但是父节点现在是数据类型，最终叶子是目的(和以前一样)。列出的候选目标是"LOCATION"节点的叶子节点。我们还将数据类型组织成四组:PHONE身份证、电话状态、个人数据和传感器。

数据类型推断

分类法的层次结构允许我们通过首先推断数据类型来显著缩小候选目标。本节描述MobiPurpose如何预测"myLat:40.44;myLon:-79.94"这个位置。

数据类型推理在网络流量中得到了广泛的研究。过去的项目要么使用硬编码正则表达式，要么使用监督机器学习方法来分类数据类型。

然而，正则表达式不能推广到不可见的模式，并且为我们的分类法标记训练数据将是一项艰巨的任务。相反，我们采用传统的生成式NLP技术来避免数据标记，并将自动化扩展到细粒度数据类型。更具体地说，我们首先使用bootstrapping方法来构建一个键值文本模式语料库，然后从语料库中派生一个贝叶斯概率模型来对数据类型进行分类。

使用模式引导来构建语料库

我们的bootstrapping方法是受到信息提取任务中先前工作的启发，这些任务需要最少的人力参与。关键思想是利用键值文本模式冗余。我们注意到，数据集中的许多键值对都是受约束的文本模式的组合，原因有两个。首先，开发人员共享类似的命名约定，这限制了键名的变化。例如，纬度键名的典型模式是:“lat”、“***_lat”、“lati”、“latitude”等。其次，我们的设备也共享类似的配置(例如物理位置、设备模型)，这也限制了值的变化。如果开发人员试图收集一些GPS数据，这些设备将向服务器发送几乎相同的值。

更具体地说，如果我们手动从几个位置键名模式开始，那么我们可以找到更多包含未知值模式的键值对。然后，可以使用这些新的值模式再次查找更多的键值对，这次是使用新的键名模式。在每次迭代中，算法只保留最可靠的迭代。这个迭代过程可以用最少的人力收集大型文本模式语料库。

Key: Value	bag-of-words	special string
devid: 99349319-a6c7-4657-a3bc-6929c52090e1	dev, id	UUID
ip_addr: 128.237.175.242	ip, addr	IP
device_model: Nexus_6P	device, model, nexus, 6p	
pickup_lat: 40.4431531	pickup, lat	LATITUDE_NUM

提取文本模式。 简单解决方案是使用精确的字符串匹配作为文本模式。但是，它不太适合开发人员变量上下文。例如，常见的纬度键名包括“dev_lat”，“device_lat”、“my_lat”、“mylat”等等。uuid的值是普遍惟一的，但是共享一个公共的字符串模式。我们需要一个更通用的表示来捕获这些重复的文本模式。通过实验，我们提取了任意键/值字符串的两种文本模式:

- 特殊字符串检查:首先使用正则表达式和双字母模型来识别常见的特殊字符串，如MAC地址、IP、电子邮件、URL地址、UUID、广告id、MD5散列、包名、时间戳、isnumber、纬度/经度、开发人员版本号、随机生成的字符串等。
- 单词包:如果键/值字符串不是一个特殊的字符串，我们使用开放源码的英语单词分割模型将该字符串解析为单词包表示。为了更好地处理技术术语(如lte、gsm)和临时缩写(如ad for advertising)，我们手动将这些通用术语添加到开放源码模型中。

Bootstraping文本模式。 算法显示了引导过程的伪代码。使用一些手动种子规则初始化方法，以提供学习的初始规则。例如，如果键包含" lat "或" lon "值是一个合法的经纬度字符串，数据类型为LOCATION。

我们

(1)使用初始规则来找到一个初始位置键值设置(表示L-KV)

(2)提取的所有文本模式出现在L-KV集

(3)计算的频率不同的文本模式和识别文本模式通常发生(如果他们的频率超过一个阈值)

(4)使用这些文本模式搜索整个数据集识别更多的位置键值对

(5)更新L-KV集，然后返回(2)。

该过程重复执行，每次以较高的频率阈值确保高可靠性，直到位置KV集合收敛

识别初始种子集是bootstrap算法的必要的第一步。如果仔细调整频率阈值，选择不同的种子只会对结果产生轻微的影响。在我们的实现中，每个数据类型的初始规则集平均包含7.5个种子规则。设备的数据类型种子规则最多，文本模式多样，IP的数据类型种子规则最少。

ALGORITHM 1: MobiPurpose's bootstrapping algorithm for data type inference

Input: A large collection of key-value pairs R and a list of data types $D = \{d_1, d_2, ...\}$
Define: For any key/value string, we extract text patterns $T = \{t_1, t_2, ...\}$.
Initialization: For each data type d_i in D, we initialize a set of seed rules to extract a initial key-value set E_i.
Bootstrapping Algorithm:
Step 1: Traverse all the keys and values in E_i, and extract the patterns as T_k and T_v.
Count the frequency of unique patterns in T_{key} and T_{value}, and remove less common patterns.
Annotate the results as T'_k and T'_v.
Step 2: Search key-value pairs in R using T'_k and T'_v.
If the key matches T'_k **or** the value matches T'_v, we add that pair to E_i.
Go back to Step 1 to find new patterns, and repeat the iteration until the size of E_i does not grow.
Output: A text pattern corpus where key-value pairs are labeled with different data types.
Bayesian classifier:
Given any key-value pair, we estimate the likelihood of different data types using a bayesian method:
$P(c|k,v) = \frac{P(k,v|c)*P(c)}{P(k,v)}$
Determining the data type for that key-value pair is equivalent to finding c^* that maximizes $P(c|k,v)$:
$c^* = \arg\max_c P(c|k,v)$

贝叶斯分类

引导方法可能会遗漏每个数据类型的一些键值对。此外，运行引导迭代可能很慢。为了解决这些问题，我们使用bootstrap结果作为语料库来量化不同文本模式的权重，并构建概率贝叶斯分类器来推断数据类型。

让 C = {c1,c2, ...,cn } 为n个数据类型(在我们的分类中n = 16)，E = {E1, E2, ..., En} 是每种数据类型的聚合KV组。

给定任意键值对(k,v)，我们可以提取文本模式X = x1, x2, x3，...表示这一对。(k,v)的数据类型为c的条件概率是P (c|k,v)确定数据类型等同于查找c显示最大化P (c|k,v)根据贝叶斯规则，P (c|k,v)可表示为:

$$c^* = \arg\max_c P(c|k,v), \qquad where \qquad P(c|k,v) = \frac{P(k,v|c) * P(c)}{P(k,v)}$$

其中P (c)表示在不观察(k,v)的情况下选择c的先验概率，P (k,v|c)为似然函数，表示数据类型c中文本模式出现的概率，P (k,v)为归一化常数。

我们对P (ci)进行建模，表示每个数据类型在所有键值对中的百分比。例如，如果MobiPurpose标识1000个隐私敏感对，其中30个是位置对，P(位置)是0。03。我们使用文本模式来表示每个键值对;因此，似然函数被解释为可能性的乘积P (xi|c)c类中发生的每个文本模式:

$$P(k,v|c) \sim P(X|c) = \prod_1^n P(x_i|c)$$

数据收集目的的推断

根据数据类型预测，我们可以通过分类法查找找到一小部分相关的目标候选项，然后使用机器学习技术来推断可能的目标。为此，我们利用了来自数据标记步骤的10个参与者。具体来说，在标记之后，我们讨论了他们观察到的数据模式(DP)，并询问他们如何确定标记的目的。基于此讨论，我们从三个独立的数据源(流量请求)派生出许多计算特性、app描述和主机域信息。

通过标记参与者和特征提取观察到的数据模式

在这里，我们报告参与者为推断目的所提到的数据模式，以及它们是如何通知我们所提议的特性的。

分类模型中使用的功能可以分为三类:嵌入式文本功能、源应用程序功能和域功能。

- DP1：主机路径中的关键字(例如searchnear、fetchads)和键值对(例如accesstoken)常常直接描述API行为。几乎所有的参与者都在标注过程中积极寻找特殊的关键词，如“ad”和“analytic”。给定任意端点地址(主机+路径)，我们通过URI协议(即，方案://子域.域/路径/文档.扩展?查询#片段)，使用上节中描述的概率模型将每个组件(scheme除外)分割为单词，并将结果编码为“单词包”表示(即 URL中出现的单词数)。我们保留了一个列表，如“com”，“www”，“android”，“谷歌”，以过滤掉一些不必要的功能。我们还对键值对应用相同的方法，并将结果合并到一个“单词包”表示形式中。
- DP2：应用程序包名和域名之间的字符串相似性可以指示应用程序是否连接到第三方服务。例如，应用程序“net.passone.gwabangeng”接触“api.passone.net/ ggwabang/questionapi .php”，表示端点不是第三方服务。我们将端点地址分割为三个部分(子域、域、路径)，并分别计算包名和三个组件之间最长的公共子字符串。然后我们用一个向量来描述这个数据模式，如果相应组件包含一个长度超过四个字符的非停止字公共字符串，则表示向量中的每个项。
- DP3：共同发送的数据可以帮助确定目的。例如，广告服务经常收集ID和屏幕大小，因为它们需要确定广告的视觉外观。分析服务通常比其他服务更积极地收集键值对(10+)。我们使用一个数值向量来表示总键值对的数量以及键值对的不同数据类型的数量。
- DP4：参与者经常阅读谷歌Play中的应用程序描述来理解网络请求上下文。在这个功能组中，我们只在谷歌Play中使用app类别。为了减少不必要的功能，我们合并了所有的游戏类别来减少功能维度。
- DP5：理解服务提供者(域所有者)业务可以帮助进行目的推断。例如，“doubleclick.com”收集的ID很可能用于广告目的，因为“DoubleClick”是提供广告服务的谷歌的子公司。参与者经常检查谷歌搜索，谁来推断业务类型。我们开发了一组脚本，使用WHOIS API2和Crunchbase API3自动化域所有者查找。我们使用Crunchbase中的企业类别来表示组织业务类型。

然而，大多数国际公司都没有Crunchbase上的简介，许多域名所有者都选择了Crunchbase“私隐注册”以隐藏他们的WHOIS注册资料。我们只能用这种方法识别22%的域名。为了解决这个问题，我们利用应用程序源代码来表示业务类型。直观的感觉是，如果一个应用程序联系了一个域，应用程序类别可以指示域所有者的业务类型。例如，我们的网络追踪数据集中有五个应用程序与uber.com联系，其中三个属于“地图和导航”类，另外两个属于“旅游和本地”类。我们可以用这两个类别来描述Uber的业务。这种方法还可以识别提供第三方服务的域，因为所有类别的应用程序都与它们联系。

我们提取这个域特性如下。我们首先使用Androguard对所有185k应用程序进行反编译，并在反编译源代码中索引域。对于每个域，我们找到源代码中包含域的应用程序，并计算不同应用程序类别的应用程序分布。我们使用两个启发式规则来提取特征表示：

1)如果域名被5+个类别的10+个app联系，我们将业务类型设置为“第三方库”

2)否则，我们使用top 3 app category来表示business type。

特征选择

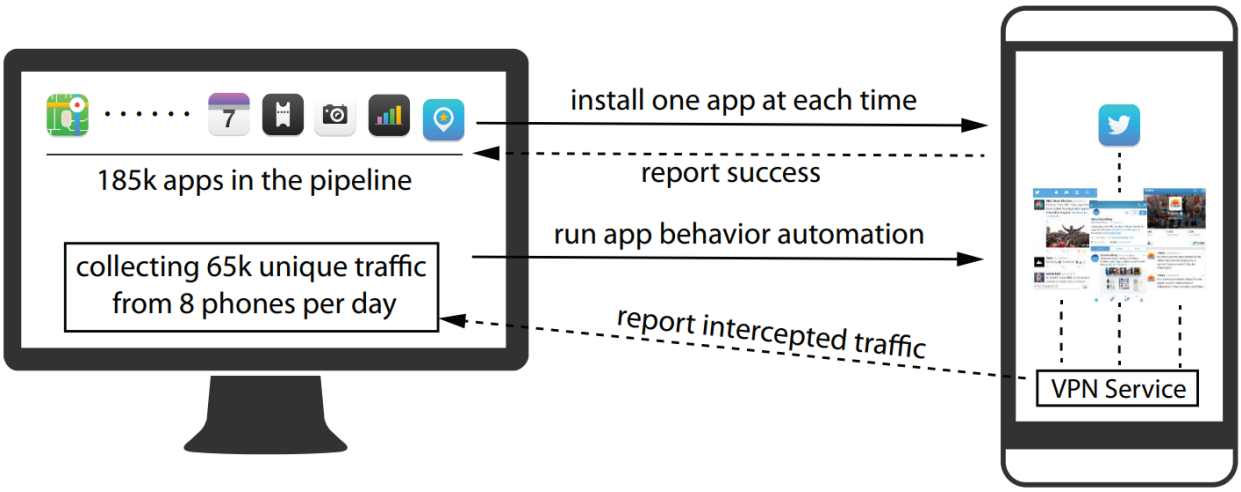
一个简单的单词袋模型产生了太多的特性，这可能导致过度拟合。我们假设低频特性不那么通用，所以我们过滤掉了文档(即流量请求)频率低于3%的所有特性，从而得到不同数据类型的特征向量，其维数在160到350之间。

监督的机器学习

我们使用监督机器学习来训练一个基于所提特征的目标分类器。我们假设异构数据源对分类过程的贡献类似，因此将所有权重组件设置为1.0。我们为每种数据类型维护一个独立的分类器，并实验了三种不同的分类算法:使用线性核支持向量机(SVM)的支持向量机、最大熵(ME)和C4.5决策树。后文我们会比较不同算法和不同特征组合的性能。

数据收集

在本节中，我们描述了我们的网络跟踪系统的设计和实现，这个网络跟踪系统的目标是使我们能够快速、方便地从实验室的大量应用程序中收集各种网络数据。

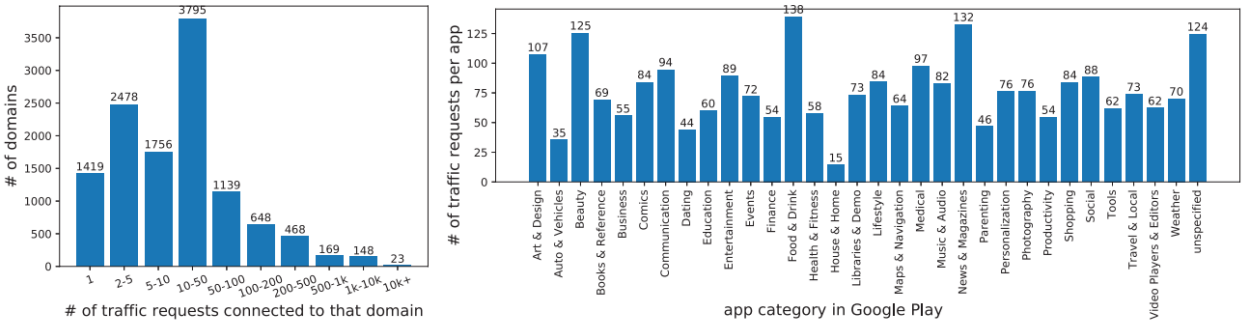


我们的跟踪系统由两个主要组件组成:一个UI自动化测试工具和一个网络嗅探应用程序。我们使用DroidBot实现了UI自动化工具，DroidBot是一个轻量级的UI引导测试输入生成器，它实时分析用户界面并随机遍历屏幕上的UI元素。然后，我们使用Android提供的内部虚拟专用网(VPN)服务构建了网络嗅探应用程序操作系统。我们手动向测试设备添加了一个受信任的根证书，并执行了一个中间人(Man-in-the-Middle, MITM)用于解密SSL/TLS通信的SSL注入。我们的架构使我们更容易捕获哪个应用程序发起了网络请求，如果我们使用服务器代理方法，这将更加困难。对于每个流量请求，我们记录目标域、路径、源应用程序和网络请求体。

我们使用这个跟踪系统测试了大量Android应用程序。我们爬过谷歌Play的网页，创建了一个所有美国用户可见的应用程序的索引，其中150万是免费应用程序。我们决定只关注2015年后更新的免费应用，结果是185,173个应用。对于每个应用程序，我们的自动化工具首先研究应用程序行为3分钟，然后卸载应用程序。3分钟的持续时间是我们的计算资源和收集尽可能多的独特流量请求的目标之间的权衡。我们的硬件配置由1台PC和8台Android手机组成。粗略地说，使用这个设置，我们每天可以拦截大约65k个网络请求和3k个应用程序。

描述性统计。 我们的数据收集过程花了50天时间遍历185k个应用程序。由于操作系统兼容性(我们的设备运行在Android 7上)，我们只能安装30,075个应用程序。我们总共拦截了14,910个应用程序的2,008,912个流量请求，联系了12,046个惟一域(302,893个惟一端点url)，向远程服务器发送了6,376,833个键值对数据。前三位活跃域名分别是doubleclick.net(261048个请求)、googlesyndication.com(158672个请求)和startappservice.com(124289个请求)。跨域请求数服从长尾分布:仅通过一个app联系9320个域(77.3%)，联系5653个域(46.9%)不到10次。

预处理数据集。 如果我们直接研究单个请求，最终的系统将偏向于顶级域，如doubleclick.net，并且不太适合处理大量的流量请求。因此，我们首先过滤掉发送空数据体的流量请求，然后合并重复的流量请求。我们使用手工编写的正则表达式根据端点地址(即主机+路径)对请求进行分组，得到60,753个惟一网络API的痕迹。例如，我们使用graph.facebook.com\/v\d.\d\/\d{15}来对发送到facebook graph API的类似流量请求进行分组，并合并它们的参数(即键值对)。



标识请求的真实行为

我们的数据标记目标有三方面:

1)评估数据类型分类的准确性

2)测试分类的完整性

3)准备用于目的推断的数据集

我们将每个标记任务设计为一组关于特定流量请求的简短问题，重点放在一个键值对上。我们向参与者展示原始流量请求以及数据类型分类结果。然后让他们判断分类是否正确，并标注数据收集的目的。

在测试中，我们发现判断数据类型对于工程师来说是很简单的，而标注数据收集的目的则需要更多的上下文知识。为了适应这种情况，我们引入了多个快捷方式(例如谷歌Play、WhoIS)来帮助参与者快速访问相关的外部信息.

过程。 为了避免测试数据泄漏，我们从预处理数据集中随机抽取了5k个API跟踪进行评估。我们开发了基于剩余56k API跟踪的引导算法，并在5k保留的API跟踪上运行最终的数据类型推断算法。标记实例来自5k保留的API跟踪。我们尝试为每个目的手工标记至少30个实例。然而，在我们的流量数据集中，一些purpose实例非常稀少(<5%)。例如，在250个位置实例中，我们只观察到5个“反向地理编码”实例。我们在为公共目的获得30个实例之后就停止了(>=5%)。

参与者。 我们招募了10名有Android应用开发经验的工科研究生。所有参与者都有超过三年的Android/iOS开发经验，其中6人是计算机科学博士生，4人熟悉逆向工程和移动隐私研究。参与者也知道实验室的GPS位置，因此他们可以识别出协调的集合是否是附近的位置。

我们在5分钟的标记教程中介绍了我们的分类法。由于目的解释可能是主观的，在某些情况下是模糊的(例如，由于信息不足或不完整)，我们为每个任务收集了三个独立的标签，并允许参与者在需要时为每个实体标记多个目的。目的标记是非常耗时的。每个标记任务需要30秒到2分钟，因为参与者通常需要生成搜索查询并在外部网站上检查事实。

标签数据统计: 我们从3177 (1059 x 3)个标签任务中收集标签，其中每个实体由三个独立的人工专家进行标签，涵盖7种数据类型和34种数据用途。结果有一个非常高的注释器间一致性：只有23个流量请求(23/1059 <2%)收到了不一致的注释，这让我们相信标签的质量是高的，并且基本上是一致的。

目的标记的结果进一步说明了从app的嵌入属性和外部因素(如接触位置)推断数据收集目的的可行性。在标注任务中，如果发现没有足够的信息支持自己的判断，参与者可以选择标注“信息不足”(II)。123项(123/3177 < 4%)标记任务被标记为II类，其中3名参与者仅标记3个实体，2名参与者标记20个实体。

我们使用多数票将不同参与者的目的标签合并在一起。如果至少有两个独立的参与者相信键值对与特定的目的相关联，那么我们接受这个标签。根据这个标准，有118个交通请求(118/1059 < 12%)太模糊而无法达成协议。在后面的目的推断评估中，我们排除了这些流量请求。

分类法实践： 如果参与者发现当前分类法没有涵盖行为，那么标签接口还允许他们添加新的用途标签。我们的参与者在30个标记任务中指定了新的目的(1.2%)，重新编码后产生了4个新类别:"网络信息-广告"、"位置-反向"、地理编码"，"位置-恶意"，"ID -恶意"。我们将前两个目的合并到最终分类中，并单独讨论恶意目的。

根据标记任务，我们认为当前的分类法对大多数常规网络流量具有良好的覆盖率。如果需要添加/更新分类法，可以不时重复上述过程。

评估

MobiPurpose使用两个步骤来推断目的。MobiPurpose首先推断数据类型，使用它通过分类法查找找到相关的候选用途。然后，MobiPurpose使用监督机器学习的方法来预测数据收集的目的。在这里，我们对数据类型推理和目的推理进行了评估，发现MobiPurpose对数据类型推理的平均准确率为95%，对数据目的推理的平均准确率为84%。

相关工作

MobiPurpose主要涉及描述移动数据访问和公开的三种主要方法:静态分析、动态分析和网络分析。

局限性

网络跟踪

从移动应用程序中收集大规模的综合数据是一项具有挑战性的任务。本文使用的UI monkey方法可以用相对较小的计算资源收集到较大的数据集：然而，它也存在着缺乏全面性的问题。例如，文本输入框验证(例如，登录屏幕)可以通过应用程序阻碍monkey的进程。在我们的实验中，我们也发现monkey无法在一些游戏应用程序中解析Unity4接口，因为很难解析UI元素树。

为了缓解这些问题，我们已经实施了几种启发式的方法。

首先，我们手动登录到流行的应用程序(如Yelp、Twitter、谷歌)，以避免这些应用程序中的登录屏幕。

其次，我们检测第三方登录选项(例如，用您的谷歌帐户登录)，并在可用时选择这些选项。最近的一项研究表明，超过40%的移动登录屏幕支持第三方登录功能。

第三，如果应用程序界面是在Unity中编写的，我们的猴子会随机点击不同的坐标。

应用程序需要登录

为了进一步了解登录限制，我们研究了登录阻塞对bot交互的影响程度。我们手动安装了美国Android商店中排名前80的免费应用程序，发现其中41个支持登录，22个支持谷歌/Facebook/Twitter登录。在剩下的19个应用中，有15个应用(例如Chase Mobile、Robinhood)需要关键的登录才能与核心功能交互。大多数关键的登录应用程序出现在前40名，而只有2个40-80岁的应用程序需要关键的登录。现代移动应用程序设计指南常常建议，注册是采用的

一个障碍。过早强制注册会导致85%以上的用户放弃该产品。我们预计，需要在“排名较低”的应用程序中进行关键登录的应用程序将会减少。

从技术角度来看，使用虚假概要文件的自动登录可能是一个法律灰色地带，这可能会引发两个法律问题:

1)接受条款和条件时进行数据包嗅探

2)使用虚假概要文件登录。

首先，登录到服务通常需要接受条款和条件，这可能会禁止包嗅探分析。禁止数据包嗅探出现在早期的最终用户许可协议(EULA)中，并受到了强烈的反对。最近的EULA模板和主要应用程序使用的服务条款(如谷歌)只禁止源代码反向工程。其次，在web/移动数据抓取中，大量使用了假配置文件登录。最近，美国一家法院裁定，根据美国宪法第一修正案，创建虚假个人资料可以受到保护。今后的工作也应注意这些潜在的法律影响。

混淆

我们的前提是，移动网络流量文本数据、应用程序源和域信息反映了底层开发人员的目的。然而，开发人员可能有意或无意地混淆了名称。例如，我们注意到一些应用程序的键名不清楚，比如“v2”、“c12”，这些键名可能在不同的流量请求之间不一致。我们在12,046(1.8%)个域名中找到218个，在14,910个域名中找到405个(2.7%)应用程序存在某种形式的混淆。

证书锁定

我们当前的MITM SSL注入实现不能拦截固定的证书通信。使用在系统级禁用SSL证书检查的修改后的操作系统可以潜在地缓解这个问题。在我们早期的实验中，我们发现这种方法只适用于使用Android SSL库开发的应用程序。所以为了更好的兼容性，我们决定使用原来的操作系统。为了量化证书锁定问题，我们选取了2018年3月13日检索到的谷歌Play免费应用前500名，发现所有网络请求中只有22个应用(<5%)使用了证书锁定。

讨论

理论框架：上下文完整性

我们的工作可以被认为有助于促进隐私作为上下文完整性的日益增长的机构[53]，它认为数据访问或披露是合法的，基于特定的上下文和规范，在其中发生的信息流。2012年，白宫在《隐私法案》中进一步支持这一框架：“消费者有权期望公司收集、使用和披露个人数据的方式与消费者提供数据的环境相一致。”

上下文完整性背后的一些思想已经开始扩散到移动计算的研究中(有意或通过聚合进化)。一个例子是app权限系统。研究人员也开始研究如何结合用户的隐私期望。例如,WHYPER和AutoCog构建了一个机器学习模型，以确定权限的用途是否与应用程序描述一致。ASPG和CHABADA通过基于应用程序描述对类似的应用程序进行聚类，来识别异常值权限。Lin等人和Wang等人在二元异常检测之外，以人们对数据采集目的的期望的形式构建了移动隐私。

在研究理解网络流量中的“为什么”的好处方面已有丰富的文献。例如，Van Kleek等人手动标记网络流量目的和发现目的，可以帮助用户做出自信和一致的选择。与之前的工作相比，MobiPurpose是第一个可以自动进行移动通信目的推断的解决方案。

分类动机与完整性

如果数据公开是合法的还是恶意的，我们的分类不能捕获。流量请求是否合法的问题只能在产生该问题的每个特定上下文中回答。但是，试图回答这个问题是具有挑战性的，因为在定义属于隐私范畴的麻烦活动时存在混淆。我们的分类将帮助我们分析各种隐私问题，因此我们可以更好地解决它们，并在它们与相反的利益之间取得平衡。

我们创建了一个包含16种隐私敏感数据类型和76种用途的分类法，10个参与者使用该分类法标记了1059个实例。虽然我们的分类法对于我们的目的来说已经足够好了，但是可能还有我们没有发现的其他目的。此外，根据使用目的的不同，我们的分类可能过于细粒度或过于粗粒度。例如，我们可以将数据分析的目的进一步划分为开发人员分析(例如，崩溃报告)和市场分析(例如，市场归因分析)。但是，我们认为所提议的方法应该适用于新的目的和数据类型。

结论与展望

本文设计并实现了第一个能够对移动网络流量中敏感数据的采集目的进行自动分类的系统。解决方案的核心是依赖于数据类型的用途分类，在该分类中，我们枚举与每个数据类型关联的潜在用途。给定流量请求中的任意键值对，我们首先使用引导NLP方法来推断数据类型，然后使用监督机器学习方法来预测数据收集的目的。我们使用由十名人类专家交叉标记的数据集来评估我们的方法。我们的实验表明，我们的方法可以预测“什么”的平均精度为95%(在8个独特的类别中)和“为什么”的平均精度为84%(在19个独特的类别中)。

过去的研究表明，表面上数据收集目的的隐私指标可以帮助用户做出隐私决策。因此，我们的下一步是进一步扩大我们的分析，并建立一个公共资源，可以帮助开发人员、最终用户、记者和政策制定者更好地了解哪个应用程序正在收集关于我们的数据、哪些数据、数据的去向以及为什么要收集这些数据。