

Characterizing Location-based Mobile Tracking in Mobile Ad Networks

Boyang Hu¹, Qicheng Lin¹, Yao Zheng², Qiben Yan¹, Matthew Trogia², Qingyang Wang³

¹The Department of Computer Science & Engineering, University of Nebraska-Lincoln, Lincoln, NE, 68588, USA

²The Department of Electrical Engineering, University of Hawai'i at Mānoa, Honolulu, HI, 96822, USA

³The Department of Computer Science & Engineering, Louisiana State University, Baton Rouge, LA, 70803, USA

Abstract—Mobile apps nowadays are often packaged with third-party ad libraries to monetize user data. Many mobile ad networks exploit these mobile apps to extract sensitive real-time geographical data about the users for location-based targeted advertising. However, the massive collection of sensitive information by the ad networks has raised serious privacy concerns. Unfortunately, the extent and granularity of private data collection of the location-based ad networks remain obscure. In this work, we present a mobile tracking measurement study to characterize the severity and significance of location-based private data collection in mobile ad networks, by using an automated fine-grained data collection instrument running across different geographical areas. We perform extensive threat assessments for different ad networks using 1,100 popular apps running across 10 different cities. This study discovers that the number of location-based ads tend to be positively correlated with the population density of locations, ad networks' data collection behaviors differ across different locations, and most ad networks are capable of collecting precise location data. Detailed analysis further reveals the significant impact of geolocation on the tracking behavior of targeted ads, and a noteworthy security concern for advertising organizations to aggregate different types of private user data across multiple apps for a better targeted ad experience.

Index Terms—Ad networks; data privacy; location-based advertising; network traffic analysis.

I. INTRODUCTION

Digital advertising market has changed dramatically since the invention of mobile devices. According to Statista [1], while desktop ad spending remains roughly the same, mobile ad spending has grown from 1.57 million U.S. dollars in 2011 to 50.84 billion dollars in 2017. The tremendous growth is in sync with the increasing popularity of mobile devices and apps. Based on the Flurry Analytics data [2], an average U.S. consumer nowadays spends five hours a day on mobile apps. Such extended mobile screen time allows ad networks to collect detailed user profiles via ads-enabled mobile apps. In particular, mobile advertising networks have been known to collect data based on users' locations, which allows advertisers to launch targeted advertising campaigns. With a controlled ad campaign experiment, a recent study [3] shows that location-based ad campaign increases daily active users by 85%.

However, despite its popularity, the privacy implications of location-based data collection require further scrutiny. Existing works on mobile privacy have focused on longitudinal studying, i.e., how privacy leakage changes over time, using methods such as static analysis [4], dynamic analysis [5], network traffic analysis [6], and hybrid approaches [7]. But

transverse studies regarding location-based data collection, i.e., *how privacy leakage changes across locations*, have not been well investigated.

Compared to longitudinal studies, transverse studies collect data samples across a wide range of locations at a specific point in time. Such methodology poses a challenge for the large-scale measurement. To obtain unbiased results, the researchers must collect mobile traffic at multiple locations and try to eliminate "time" as an independent variable. Not only does such an experiment require a large number of devices and location samples, but it also relies on effective location spoofing to bypass the ad networks' location verifications. Naive approaches that spoof GPS signals can be easily detected by cross-referencing the GPS coordinates with the devices' network profiles or users' regular activities.

To overcome the challenges, we construct an automated fine-grained data collection instrument running across different geographical areas. We identify the *hot zones* and *cold zones* for mobile privacy study using the physical locations of the advertisers whose websites contain online tracking contents. Our intuition is that mobile tracking originates from web tracking, and therefore the advertisers engaging in web tracking also likely adopt mobile tracking. To bypass location verification, we consistently spoof the GPS coordinates, network profile, and user activities, to let them appear coherent in the eyes of ad networks.

Our instrument also allows us to understand how mobile ad networks aggregate information across apps, by running multiple apps packed with the same ad library. With the lack of tracking cookies in mobile apps, ad networks incorporate ad libraries in different mobile apps which request different sets of permissions. By linking the permission profiles of different active apps at different locations to the same ad network, we could evaluate the extent to which the ad networks fuse users' information for targeted advertising.

Based on our extensive measurement with real-world apps, we make a number of interesting discoveries: 1) mobile web ads and mobile in-app ads contact a similar set of popular third-party domains; 2) although the mobile ad network traffic are relatively secure, the low adoption of HTTPS at the advertisers' side still lead to the leakage of private information; 3) different ad networks present different private information collection behaviors across different locations, some of which reveal special interests in collecting particular types of private information; 4) most ad networks can infer users' precise lo-

cations even without collecting fine-grained GPS coordinates. This paper makes the following contributions.

- We design an efficient privacy leakage measurement system to characterize fine-grained location-based mobile tracking. The system can adjust the GPS locations and network profiles, conduct traffic collection, and perform detailed traffic analysis.
- We develop novel domain classification mechanisms to accurately classify the collected domains into ad network domains, advertiser domains, and location-based ad domains.
- We identify the private data collection behaviors of ad networks at the organization level. We find that there is an alarmingly comprehensive set of users' private information that the ad network organizations can collect by aggregating data from multiple apps.
- We expose ad networks' information collection behaviors across different locations. Our findings suggest that ad networks manifest different private information collection behaviors at different locations. Location leakage by ad networks is particularly disconcerting, as most ad networks can either collect or infer precise locations.

II. RELATED WORK

The existing privacy research on the mobile ad networks mainly focuses on the malicious uses of advertising contents, which include malicious adSDKs and malicious ad creative. Earlier studies suggested that adSDKs often have poor security and exhibit fraudulent behaviors [8]. Researchers have raised concerns of malicious advertisers recently [9], [10], who can obscure the apps' background to hide malicious activities. In response, ad networks are rapidly improving their screening process to filter out malicious ads and require a minimum number of targeted audiences to prevent individual targeting [10].

Demetriou et al. [11] present the first measurement system to reveal the potential risk of ad libraries in mobile apps. Recently, researchers have discovered that the third-party ad libraries in mobile apps misuse their inherited permission and access rights to learn and track users' private information without explicit consent [6], [7]. Both static and dynamic analyses tools have been developed to detect privacy leakage in mobile apps.

Static analysis approaches. Static analysis is largely scalable and has a low overhead to perform, and it identifies potential privacy leakage through application code analysis. Static analysis of application binaries has been used to detect malicious data flows [4], malware classification [12], and user activity analysis [13]. The changes across different versions of ad libraries [14] have made the mobile systems more vulnerable because of the adjustments in permission requests across platform/app versions.

Dynamic analysis approaches. Existing studies have provided useful tools to identify the misuse of privacy data through dynamic tainting analysis [5]. The location leakage through location-based services (LBS) has been analyzed [15], [16]. In this paper, we analyze apps across different cities

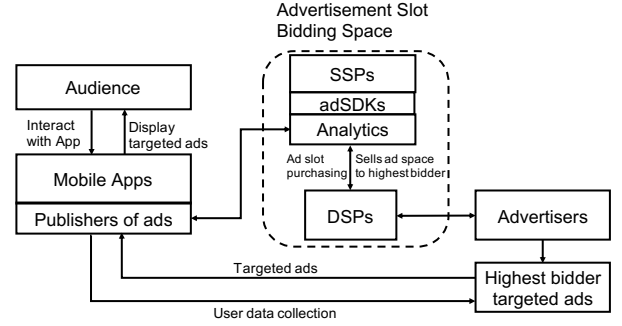


Fig. 1: Mobile advertisement ecosystem.

in United States to understand the behaviors of mobile ad networks across different locations. We also consider cross-application privacy leakage by aggregating the collected private information from the same ad domain across multiple apps. For improving the coverage of dynamic analysis, researchers have developed “UI Monkeys” to automate the input generation. Customizable tools like Android Studio’s Monkey¹ and Appium² allow researchers to provide a customized simulation of app interactions.

III. BACKGROUND

A. Mobile Advertisement Ecosystem

The digital advertising ecosystem consists of four types of entities: audiences and publishers, sell-side platforms (SSPs), demand-side platforms (DSPs), and advertisers, as shown in Fig. 1. Audiences are the users who watch the ads when they interact with the contents of a publisher. Publishers are the owners of websites or apps that serve ads, which include SSP toolkits, such as analytic scripts and advertising libraries (i.e., adSDKs for Mobile). DSPs facilitate purchasing ad slots and serving ads on behalf of an advertiser. SSPs facilitate selling the ad spaces to the highest bidder in a publisher’s content by auctioning them to DSPs. Advertisers are entities that have ads to display. Advertisers may upload the actual ad content, known as ad creatives, to a DSP, or host them on their servers and provide URLs for the DSP to display.

In the web ad environment, the third party cookie has been the universal tool for tracking host information to provide targeted ads. Any website that uses the ad domain can access the cookie of this particular ad domain, which allows for cross-site targeted advertising. In contrast, mobile in-app ad environment does not use shared cookies for tracking. Instead, the mobile advertisement ecosystem relies on application stimulus, which collects private data protected by permissions. Our analysis includes a detailed inspection of the tracking data that comes through as macro parameters in the Uniform Resource Locator (URL) of the network communications from mobile devices.

Apparently, the more information SSPs can provide to the bidders, the higher bids they will get. Therefore, SSPs are motivated to collect a variety of information, such as: mobile

¹<https://developer.android.com/studio/test/monkey>

²<http://appium.io/docs/en/about-appium/intro/>

TABLE I: List of PII categories and types

Unique Identifier	Advertising ID, Android ID (device ID), hardware serial, IMEI, IMSI, MAC address
Personal Information	data of birth (DOB), email address, first and last name, gender
Location-related	GPS location, IP address, zip code
User Credentials	username, password

TABLE II: Supported location granularity of top 30 mobile ad networks

Supported finest location granularity	# of ad networks
Up to country level	7
Up to city and business address level	15
Up to zip code level	4
Precise Address level	4

advertising identifiers (MAIDs), locations, network profiles, device types, etc.

B. Problem Definition

The goal of this research is to gain insights into the different privacy leakage behaviors of multiple ad libraries across different apps, organizations, and locations, and evaluate if the cross-application ad libraries can correlate the multiple instances of leaked private information for more precise ad targeting. We combine and analyze traffic from different domains that belong to the same organization to achieve a more accurate estimation of collected information by these organizations.

Personally identifiable information (PII) has been defined by NIST in 2010 as “any information that can be used to distinguish or trace an individual’s identity”. Such information is often collected by the third-party services or ad networks without users’ consent. Leveraging existing studies [6], [7], [17], we summarize a PII list containing 15 elements. We categorize these private elements into four categories, including: (1) unique identifier, (2) personal information, (3) location information, and (4) user credentials, as listed in Table I. As shown in Table II, among the top 30 mobile ad networks we surveyed, 23 ad networks provide fine-grained location-based targeted ads tailored for different cities, zip codes, or precise addresses, within which only 4 ad networks provide targeted ads for precise addresses. Therefore, our measurement switches across *different cities* for studying location-based mobile tracking.

C. Threat Model

We define three main threats that induce users’ PII leaks for mobile ad networks.

Threat from an organization with multiple domains. Popular ad networks usually contain multiple third-party services to aggregate more comprehensive private information from different domains. Thus, the ad networks are able to collect users’ private information across multiple apps. The organization-level privacy leakage study is of utmost importance to understand the power of these organizations.

Threat from adware. Some app developers may collect sensitive information via ad network libraries or other third-party services either directly or indirectly. It is difficult to tell

whether such collection is necessary for the app’s functionality. Specifically, adware has been designed to actively collect private information to serve more ads.

Threat from network eavesdroppers. Network eavesdroppers may get private information by monitoring the network communications. Some of the private information may be leaked in plaintext via HTTP. In our study, we try to evaluate the severity of such privacy leakage and understand what information an eavesdropper can obtain.

IV. LOCATION-BASED MEASUREMENT PLATFORM

Our measurement platform mainly consists of two components: location-based traffic measurement and traffic analysis, as shown in Fig. 2. In-app advertising and mobile web advertising both have their advantages and limitations in the eyes of advertisers. According to eMarketer [18], mobile apps account for nearly 86% of time spent using smartphones. But a few top apps dominate the app usage. Meanwhile, mobile web advertising may have less usage time, but there are more websites than apps on the market. Some large publisher either do not have apps or their customers tend to use websites more. Thus, in-app and mobile web advertising are both popular in today’s mobile advertising ecosystem, which guide the design of our traffic measurement system.

A. Traffic Measurement

Our traffic measurement consists of mobile devices, a wireless router, and a workstation. Mitmproxy [19] is used to intercept the traffic generated by mobile apps. We install Mitmproxy certificate on the mobile device to decrypt the HTTPs traffic. We also use Monkey, a popular input generation tool used extensively [7], [17], to automate the app interaction by randomly injecting user event sequences. We let Monkey interact with each app for five minutes in order to generate enough traffic for analysis. Many apps require users to log in with a username and password. To avoid excessive manual efforts, we record and replay the login events using RERAN [20] for such apps.

All the traffic between the app and its contacted server would go through the Mitmproxy and the router, where the traffic is intercepted and logged. Mitmproxy is capable of performing TLS interception to record the plaintext of HTTPs requests. For apps that prevent TLS interception using SSL pinning, we use JustTrustMe [21] to pass certification verification using SSL unpinning technique.

The location-based study requires a system to generate genuine location information for large-scale measurement. We use Inspeckage module in Xposed framework to change the locations. To automate the location change, we use Selenium to automatically change the GPS locations through Inspeckage’s web interface. Many ad networks cross reference the GPS coordinates with the device’s IP address. Therefore, We set up a VPN service using ExpressVPN to fake the IP addresses, which are configured to match the faked GPS locations.

As for the study of mobile web tracking, we aim to identify advertisers engaging in location-based ads. Thus, we query the

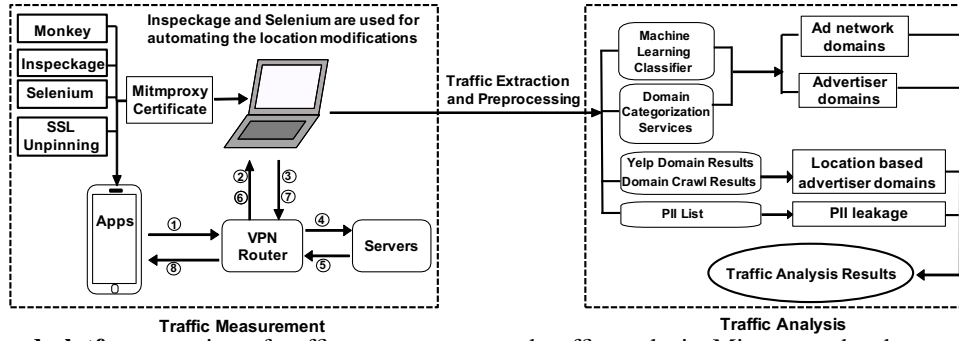


Fig. 2: **The proposed platform** consists of traffic measurement and traffic analysis. Mitmproxy has been set up as a transparent HTTPS proxy. ① The mobile client initiates a connection to the server. ② The router redirects the connection to mitmproxy. ③ mitmproxy dynamically generates certificates for the connected hosts and signs it with its own certificate. ④ The mitmproxy connects to the server via router, and establishes a TLS connection. ⑤ The server with the matched certificate responds to the client. ⑥ ⑦ ⑧ The router will redirect the response to the mitmproxy, and then forward it to the client.

Yelp Fusion API and select local businesses in different cities whose websites support mobile browsers. We use the proxy service Crawlera to query the websites with fake mobile user agents, and record sites that return no user agent errors.

B. Traffic Analysis

During traffic analysis, we focus on identifying and extracting information from network traffic related to the ad networks. We propose a comprehensive domain classification mechanism to extract the third-party domains, ad network domains, advertise domains, and location-based ad domains.

Third-party domain identification. Domains can be classified as first-party domains and third-party domains, and the owner of first-party domains are the app’s owner. We propose an empirical method utilizing domain counts to identify third-party domains, based on the observation that third-party domains appear more frequently across multiple apps than first-party domains. Specifically, for each app in our dataset, we first extract the developer information in the app’s webpage. Then, we identify the maximal number of apps that have been developed by the same developer, which we assume will use the same first-party domains. This number is defined as the threshold for identifying the third-party domains. After we extract all the domains in the traffic of all apps, we count the appearance of each unique domain across multiple apps. To avoid bias, we only count once if the domain appears multiple times in each app. We identify all the third-party domains, whose number of appearance is higher than the threshold (in our case, 9). As it is possible that potential third-party domains may be counted less than the threshold, we use other methods described below to help catch the missing third-party domains.

Ad network identification. We first generate a list of ad network domains using the publicly available information and two *domain organization mapping list* on GitHub [22], [23]. This list will be used to identify the ad networks appeared in the collected traffic. There are some unpopular ad networks that are not included in any lists. To identify all the possible ad network domains in our traffic, we utilize the DuckDuckGo search engine to query each domain and get corresponding descriptive information if available. Bi-grams and tri-grams of the descriptive texts are used as their features to classify

the domains into ad network domains and non-ad network domains. We construct our training and testing set using 2,000 non-ad network domains from Alexa, and 2,000 randomly picked ad network related domains from EasyList. In the end, the domain classification accuracy reaches around 70%.

To improve the classification performance, we propose to use three classification engines, i.e., VirusTotal, McAfee, and OpenDNS, to generate the domain classification result to be our ground truth. These engines are capable of categorizing the domains and quantifying their overall safety. For each unique domain we find in the traffic, we query the classification engines and get the related information such as category, subdomains, and the feedback of Whois lookup for the queried domains. If any of the engines considers the domain as an ad network domain, we add it to our ad network domain list.

We evaluate the performance using our ground-truth data with labels, and the domain classification accuracy can reach 92%. Although this list may not cover all the ad networks in the market since these engines cannot recognize all the domains, we consider it to be sufficient for our study. The ad network results contain not only all the popular ad networks in AppBrain [24], but also many small ad networks which have insignificant market shares.

Advertiser domain identification. The advertiser domains can also be observed in the traffic served by the ad networks. The advertiser domains are associated with businesses that post ads through ad networks. In order to identify advertisers, we refer to the three popular domain categorization services mentioned above. If any of these three services categorize the domain into advertisements (ad networks), application and software download, web analytics, and other web related categories, we consider them as non-advertisers. We consider all the remaining domains associated with other categories (shopping, education, travel, etc) as advertiser domains, and remove the ones that could not complete the categorization of all three engines. To further improve the accuracy of the advertiser list, we utilize Yelp API and query the top 1,000 business domains for each category (if available) at different locations. We add any domains that appeared in our Yelp results to enrich our advertiser domain list.

Algorithm 1 PII Leakage Identification Algorithm

INPUT: Predefined PII list (according to Table I), Domain organization mapping list.
OUTPUT: PII leakage of each app

```

1: for each App do
2:   for each location do
3:     Extract Gets and Posts URLs from captured traffic flows
4:     Extract key-value pairs from the URLs
5:     Match the key-value pairs with hashed PII values in PII list
6:     if find a match then
7:       Log the key-value pair as a PII leakage for the app
8:       Extract domains associated with the key-value pair
9:       Match domains to the domain organization mapping list
10:      if find a match then
11:        Log the key-value pair as a PII leakage for the matched organization
12:      else
13:        Log the key-value pair as a PII leakage for "Others" organization
14:      end if
15:    end if
16:  end for
17: end for
18: Return PII leakage results

```

Location-based ad domain identification. After classifying the advertisers' domains, we move on to identify whether the advertiser's related ads are location-based ads. Our goal is to associate the location of the advertiser domain with the location of the served ads. For example, if the ads appear in the area served by the advertiser, it is considered as location-based. The service area of the advertiser can be derived by querying Yelp. But simply relying on Yelp's query results may not be sufficient in identifying the local businesses. To differentiate between the local businesses and all other advertisers, we crawl all the advertiser domains and check if the front page of each domain contains the city name where the served ads appear. By combining the yelp local business list and web crawling result, we can identify the location-based ad domains.

PII leakage identification. Mitmproxy provides a standard method of reading and parsing the captured traffic. We use Mitmproxy to extract the information from the traffic flows including the domains and any PIIs. The leakage identification algorithm is presented in Algorithm 1. We first extract the HTTP/HTTPS request URL, response URL, and request/response contents. By integrating the domain organization mapping lists mentioned above [22], [23], we generate a complete leakage parameter dictionary for every organization. Then, we look up the leakage parameter dictionary to identify the known PIIs values (including hashed values with MD5, SHA1, SHA256, and SHA512) and evaluate the severity of ad networks' PII leakage at different levels including app-level and organization-level across different locations.

V. MEASUREMENT RESULTS AND ANALYSIS

In this section, we present our measurement results based on extensive experiments. We first compare the mobile web ad tracking and in-app ad tracking behaviors. Then, we expose the organization-level cross-app privacy leakage based on the traffic analysis results. Finally, we study the ad networks' data collection behaviors across different locations, i.e., different cities, rural/urban areas. We use 8 Moto G4 mobile devices with the Android 4.4.4 (compatible with JustTrustMe) or Android 7.1.2 framework to automatically launch traffic

measurement and analysis. For apps that fail to run on Android 4.4.4, we rerun them on Android 7.1.2 without SSL unpinning.

A. Measurement Dataset

We have collected two traffic datasets to facilitate the measurement study. Dataset_1 contains traffic from 1,100 popular apps running at two locations (i.e., Lincoln, Nebraska and New York City), while Dataset_2 contains the traffic from 110 apps (randomly selected from the 1,100 apps of Dataset_1) running across 10 different locations, detailed in Table VIII. Within these two datasets, we removed the apps that fail to generate network traffic in all the locations. In the end, we collect 63.0 GB traffic data: Dataset_1 contains 814,117 traffic flows from 1,026 apps across 2 locations, and Dataset_2 contains 535,655 traffic flows from 110 apps across 10 locations.

B. Mobile Web Ad Tracking vs. In-app Ad Tracking

Mobile web ad tracking allows ad networks to collect users' private information during web browsing activities. We collect the HTTP request/response URLs related to mobile web ad tracking and compare them against in-app ad tracking results. **Finding 1: mobile web ads and in-app ads contact a similar set of popular third-party domains.** For both types of ad tracking, *googleapis.com* is the most popular third-party domain. Despite such similarities, we also find some third-party domains (especially these ad network domains) only exist in the mobile traffic for in-app ads, such as *flurry.com*, *unity3d.com*, *applovin.com*, *mopub.com*, etc. The reason is that: different from in-app ad tracking that tracks both ad networks' and advertisers' domains, web ad tracking only tracks the advertisers' domains.

Finding 2: mobile web ads have a significantly lower adoption rate of HTTP/HTTPS than mobile in-app ads. We also compare the total percentage of HTTP/HTTPS traffic flows originated from domains related to web and in-app location-based ads. As shown in Table III, we can see that HTTP traffic dominates in the mobile web ad traffic. The reason is that many landing pages contain third-party HTTP content, which can cause mixed-content errors if the landing pages upgrade to HTTPS. The low adoption rate of HTTPS in mobile web ads is likely to continue as long as third parties continue to use HTTP by default. On the other hand, mobile in-app ads mostly carry HTTPS traffic. The reason is that in-app ads do not use HTTP referrer headers to indicate the sources of the redirected traffic, and thus will not incur mixed-content errors. Without such legacy issue, mobile in-app ads tend to adopt HTTPS for secure app-server communications.

TABLE III: Comparison of HTTP/HTTPS traffic from web/in-app advertiser domains

HTTP traffic (web/in-app)	HTTPS traffic (web/in-app)
84.8% / 18.48%	15.2% / 81.52%

Finding 3: mobile web ads request location via landing URLs leading to privacy leakage concerns. In our web ad traffic, we discover a significant amount of advertisers

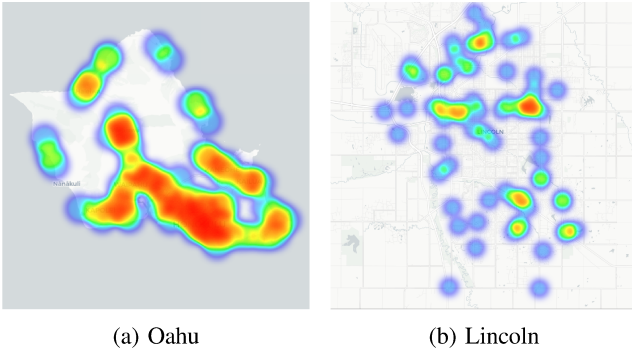


Fig. 3: Heatmap of location requests in landing URLs on the island of Oahu and Lincoln.

TABLE IV: Number of unique domains identified in each category

Dataset	All domains	Top-level domains	Ad domains	Location-based ad domains
Dataset_1	7,208	2,532	970	208
Dataset_2	4,398	1,760	539	141

who seek location information via the landing URLs' macro parameters, without explicitly expressing their purposes. These reckless behaviors allow eavesdroppers to collect and infer sensitive information about users by observing the ad traffic passing through the network. Fig. 3 shows the heatmap of location requests (with hot zones and cold zones) in landing URLs of mobile web ads on the island of Oahu and Lincoln. We can see that the location requests tend to be positively related to the population distribution. These observations motivate us to further investigate the privacy leakage through mobile ad networks across different organizations, as well as various locations with an emphasis on the hot zones.

C. Organization-level Cross-app Privacy Leakage

The advertising organizations usually own multiple ad networks, and it is conceivable that they tend to aggregate data from these ad networks to achieve a better user profiling for targeted ad delivery. In this section, we expose the organization-level data collection behaviors of popular ad networks. We also identify the different organization-level privacy leakage behaviors across different locations.

AppBrain [24] provides a list of the ad network popularity based on the number of installs of related apps. Similarly, we rank these ad networks based on the collected network traffic of Dataset_1. The result indicates that AdMob (i.e., Google ad network) is observed in the traffic flows of 601 (i.e., 58.58%) app, demonstrating the wild popularity of Google's ad network. Moreover, Unity 3d, ranked second, is observed in 180 (17.54%) apps' traffic flows. This result is consistent with the ad network popularity list of AppBrain website.

Table IV presents the number of domains identified in each domain category. Within all the domains in Dataset_1 (or Dataset_2), we identify 2,532 (or 1,760) unique top-level domains by combining multiple sub-domains. These domains belong to 496 (or 247) different organizations. Using the domain classification methods in Section IV-B, we can identify

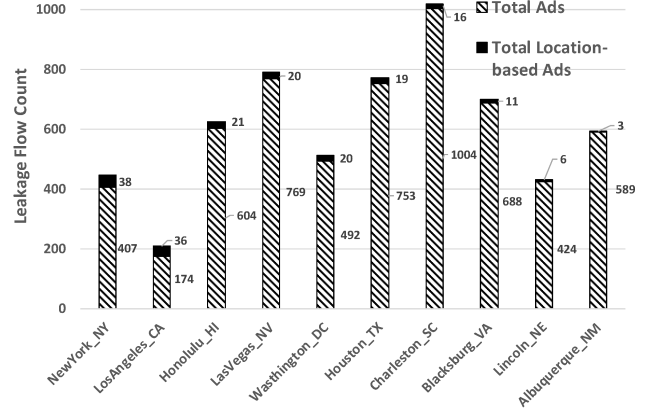


Fig. 4: Number of total ads and location based ads in each city (Dataset_2).

TABLE V: Top 10 ad organization list of PII leakage severity in all locations sorted by average PII leakage flow count per app.

Organization	Average # of PII Leakage per app	# of PII Leakage Types
LKQD	3,399	3
AOL	360	3
Facebook	322	11
SpotX	279	6
Tapjoy	211	9
Heyzap	184	6
Google	80	15
AppsFlyer	70	7
MoPub	54	2
Applovin	37	2

970 (or 539) unique ad domains and 208 (or 141) unique location-based ad domains, respectively. Location-based advertisement constitutes 21.44% (or 26.16%) of all the captured advertisements.

Finding 4: the number of location-based ads across different cities is positively correlated with the population density. Fig. 4 shows the number of location-based ads in each city based on the Dataset_2. The result shows that the number of location-based ads is positively related with the population density of the cities, which is similar to the phenomenon observed in the mobile web ad marketing environment. A similar trend is also observed with Dataset_1. In New York, we identified 782 unique advertisers and 173 unique location-based advertisers, while in Lincoln we identified 322 unique advertisers and 72 unique location-based advertisers.

We examine the app-level privacy leakage and find that Game apps are leaking private information over a large number of flows, and they leak different types of PII information. These top-ranked apps all interact with multiple ad networks (i.e., 8 ad networks on average), and the organizations behind these ad networks are able to aggregate a considerable amount of private information.

Finding 5: Popular ad networks generally collect more diverse types of PII data. For data aggregation at the organization level, Table V shows the top 10 list of the ad organizations ranked by the PII leakage severity in terms

of average leakage flow counts per app (i.e., total leakage flow counts of an ad network divided by the number of apps associated with this ad network) across all the locations. In general, the result indicates that some popular ad networks (e.g., Facebook) generate a large amount of PII leakage flows per app. A considerable number of flows from LKQD, a video ad platform (which is included in multiple apps, such as *cjvg.santabiblia* and *com.july.ndtv*), leak private information, although it only leaks three different types of PII information. The top 5 ad organizations ranked by the number of unique PII leakage types are: Google, Facebook, Amazon, ironSource, and Tapjoy. This indicates that the big companies with popular ad networks collect most types of privacy information. It is worth noting that Amazon collects 11 types of PII information, while we only find 736 flows carrying private information that are associated with Amazon ad domain within our datasets.

TABLE VI: Location related privacy leakage observed in Dataset_1

City name	GPS	IP address	Zip code	Total
New York	32,826	29,161	5,857	67,844
Lincoln	25,761	20,765	4,691	51,217

TABLE VII: The mean and standard deviation (STD) of PII leakage flow across 10 locations

PII Type	Mean	STD
Advertising ID	7,762.22	2,311.37
IP address	1,585.85	922.73
GPS	2,072.31	804.21
MAC address	778.83	751.83
Android ID	1,214.82	245.68
email	407.72	103.26
gender	112.69	50.59
IMEI	93.35	46.13
Hardware serial	13.17	4.95

TABLE VIII: PII leakage severity at each location.

City name	# of PII leakage	Ad networks with maximal collected PII
Las Vegas	22,328	AdMob
Albuquerque	17,699	LKQD
Honolulu	16,996	LKQD
Washington, D.C.	16,005	LKQD
Charleston	14,576	AdMob
Blacksburg	14,069	AdMob
Houston	13,095	AdMob
Los Angeles	11,875	LKQD
Lincoln	10,808	AdMob
New York	10,140	AdMob

TABLE IX: The PII type that is most frequently leaked by the ad networks based on 2 datasets.

PII Type	Ad Network	Collected Times
Advertising ID	LKQD	72,185
IP Address	LKQD	34,584
GPS	LKQD	28976
MAC address	Tapjoy	5,364
Android ID	Tapjoy	7,690
email	Google	4,403
gender	Appodealx	343
IMEI	Fyber	87
Hardware Series	Charboost	27

D. Location-based Private Data Collection of Ad Networks

Ad networks extensively collect users' location information. Table VI shows that the ad networks collect location information in the format of GPS, IP address, and zip code. We observe that New York has more location-related leakage compared with Lincoln. This result complies with our assumption that ad networks in larger cities will initiate more location related requests and collect more location data. The experiment with Dataset_2 presents similar phenomenon, which we omit here due to page limitation.

Before we unveil the details of ad network collection behaviors across different locations, we evaluate the difference among the leaked PIIs across different locations. Table VII shows the mean and standard deviation for the number of PII leakage flows of each PII type to measure the magnitude of the differences across 10 locations. From this table, we can see that the number of PII collections varies significantly across locations, while the Advertising ID, IP address, and GPS location are the most collected PII types for these mobile ad networks. This observation indicates that the ad networks behave differently in collecting users' private information across different locations.

Finding 6: The number of ad networks' PII leakage flows differs across different cities. To further identify the private data collection behaviors of ad networks across different locations, we extract the traffic flows related to the ad domains, measure the total number of PII leakage flows and the number of PII leakage types at each location. Table VIII shows the number of PII leakage flows vary across different locations. In addition, AdMob collects the maximal number of privacy-leaking flows within 6 cities. It is worth noting that AdMob collects the most privacy-leaking flows in almost all cities, while LKQD collects the most privacy-leaking flows in 4 cities, but it keeps quiet (i.e., collects negligible amount of privacy-leaking flows) in other cities, maybe due to its failure in the ad space bidding in these cities. Fig. 5 shows the different number of PII leakage types of ad networks across different locations. Overall, AdMob collects the most types of PIIs across all locations.

These ad networks present different behaviors across different locations, and we suspect that different ad networks may be interested in different PII types. In Table IX, we show the number of times that each ad network collects the corresponding PII information. We show the ad network with the maximal collection times, which indicates that the ad network is most interested in the corresponding PII. LKQD has the most interests in the Advertising ID, IP address, and GPS, while Google is most interested in email address.

We examine the privacy policy of all the ad networks, and we find that all the ad networks claim to collect both fine-grained location (GPS) data and coarse-grained location (IP address) data, which we have confirmed using our measurement study. Even though all the ad networks claim to collect both fine-grained and coarse-grained location data, they are still different from each other in terms of the number of

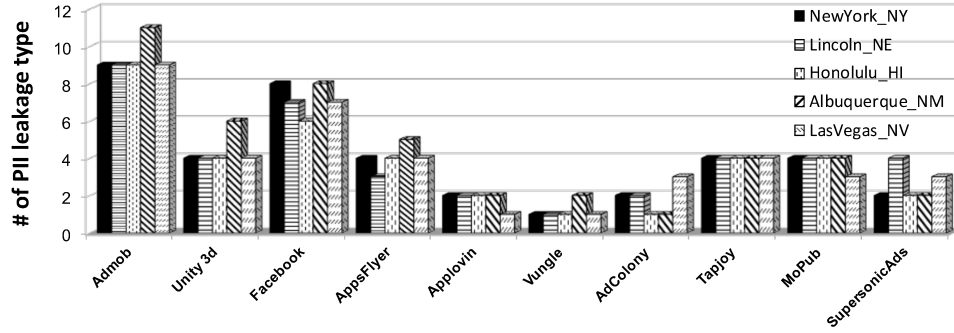


Fig. 5: The leaking PII types of top 10 ad networks across different locations.

decimals in the collected GPS location data. To put it into context, when the decimals of GPS data are 3 digits, it can be used to identify the neighborhood or street which is precise to 111.32 meters at the equator. Moreover, when it reaches to 6 digits, it can be used to identify the individuals with the precision of 111.21 millimeters at the equator.

TABLE X: The ad networks' location leakage severity.

Ad Networks	# of decimals in collected GPS	# of valid decimals
Sitescout, Mopub, Google, Appodeal, mediabrix.com, adsrvr.org, Amazon	15	8
Nexage, algovid.com, adhigh.net, LKQD, fqtag.com, AdColony	14	8
xAd, Flashtalking	13	8
Facebook, advertising.com	8	8
Irx.io, PubMatic	7	7
OpenX, Yandex, Inneractive, SpotX, Casale Media, Unity 3D, smartadserver.com, StreamRail, Smaato	6	6
Vungle, AdBuddiz, Heyzap, Applovin	3	3
Adform, Millenial Media, InMobi	1	1

Finding 7: most ad networks collect fine-grained GPS location data. Table X presents all the ad network location leakage severity. Among the 35 ad networks, 28 of them have collected user's fine-grained location (i.e., the number of decimals is greater than or equal to 6). We consider these ad networks to be aggressive in collecting precise locations since they have the ability to locate individuals very precisely. As a result, these ad networks can provide advertisers with a precise location targeting service. Moreover, among these ad networks, 7 of them collect location data with the 15 digits decimal accuracy, indicates at least 7 digits in the GPS data are useless. Potential malicious ad network or attacker can take advantage of these extra digits to embed some users' sensitive information and send to the server without getting spotted.

E. Rural Area vs. Urban Area Location-based Mobile Tracking

The aforementioned experiments prove the different tracking behaviors of mobile ad networks across different cities. As shown in Table II, most ad networks support location-based ads with respect to different cities. We set up an experiment to verify whether these ad networks behave the same at rural

area and urban area in the same city. We select 10 popular apps, which collectively include 29 ad libraries. We also pick two locations, i.e., the *Downtown* and *Lake Ray Roberts*, in a large city *Dallas* for comparison. We further randomly pick 5 points in the Downtown area, and 5 points in the Lake area. To avoid the time variability, two comparative tests (i.e., one in Downtown and one in Lake) are performed simultaneously using the same apps with the same recorded user inputs. We run each app for 10 minutes, test it 10 times at each point, and collect the network traffic.

Finding 8: A more diverse group of mobile ad networks operate in rural area, which results in more PII leakage.

We notice similar number of PII leakage for most PII types at these two different locations, while significant difference can be observed for four PII types as shown in Table XI. Generally, the rural area (i.e., Lake) collects more PII data than the urban area (i.e., Downtown), which is counterintuitive. Delving into the traffic, we notice a more diverse group of ad networks in the Lake area, who collect more PII information. Notably, LKQD and Tappx collect most of the PII information in the Lake area, but both never show up in the Downtown area. This can be attributed to the less competition in the Lake area for the ad bidding system, which brings in "less competitive" players in mobile ad business. On the other hand, the Downtown area is a highly competitive area for mobile ad networks, where "more competitive" players win with a high probability.

TABLE XI: Location related privacy leakage for rural/urban area

Location	Average # of GPS	Average # of IP	Average # of Ad ID	Average # of Android ID
Downtown	10	42	35	22
Lake	47	144	131	210

VI. DISCUSSIONS AND FUTURE WORK

SSL pinning and input automation. We investigate the private data collection behaviors of ad networks across different locations. We use real devices for our measurement study to avoid the emulation detection of some sophisticated apps. Higher version of Android system has implemented a stricter rule in preventing SSL unpinning, in which the developers can prevent traffic interception by trusting only specific/allowed certificates. As a result, we cannot decrypt HTTPs traffic from

several apps which only work with high Android version. Also, SSL unpinning does not work with all the apps, This is a common limitation of traffic analysis on Android devices. Our results are based on the traffic of the apps that could be captured. We also use Monkey to automate the user input generation, and the proposed study will benefit from the advancement of input generation tools [25] to improve the coverage.

App execution time. Our automated platform only executes one app for 5 or 10 minutes. However, in location-based advertising, the app's execution time can be a key element that impacts the traffic collection results. Some advertisers prefer to provide their location-based ads during a specific time of the day so that they can maximize the effectiveness of their ad delivery. For future study, we will record all the timestamps of our traffic and find out which period of time is the "golden" collection time for different ad networks, and how these ad networks' behaviors will change at a different time.

Traffic obfuscation. Obfuscation has been used by apps to encrypt users' private data like username, password, or email. As mentioned in Section IV-B, we hash all the known PII values with different hash functions to match traffic. However, if the malicious ad networks or attackers intentionally try to evade our analysis, they can steal the users' PII without getting spotted by using customized hash functions or encryptions.

VII. CONCLUSION

In this paper, we present a measurement study for privacy leakage in location-based mobile advertising service. We proposed and implemented a transverse measurement platform for mobile ad networks capable of location spoofing, domain classification, and privacy leakage detection. We performed extensive threat measurements and assessments with the collected traffic data. Our findings show that mobile web tracking and in-app tracking share a similar set of third-party domains, and the exceedingly high percentage of HTTP requests in mobile web ads becomes a vulnerable point inciting eavesdropping attacks. Our results verified that ad networks perform differently across different locations, and most ad networks can extract precise locations. Alarmingly, there is little correlation between ad network size and their location information leakage severity since both large and small ad networks could collect or infer fine-grained location information.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their valuable comments and feedback. This work was supported in part by the NSF grants CNS-1566388, CNS-1717898, CNS-1566443, DGE-1662487, and Louisiana Board of Regents under grant LEQSF(2015-18)-RD-A-11.

REFERENCES

- [1] "Digital advertising spending in the united states from 2011 to 2019, by channel," <https://www.statista.com/statistics/260279/digital-advertising-spending-in-the-us-by-channel/>, Accessed on Jan. 2019.
- [2] "U.S. consumers time-spent on mobile crosses 5 hours a day," <http://flurrymobile.tumblr.com/post/157921590345/us-consumers-time-spent-on-mobile-crosses-5/>, Accessed on Jan. 2019.
- [3] "Inside google marketing: How we use contextual signals to break through with mobile users," <https://www.thinkwithgoogle.com/marketing-resources/experience-design/inside-google-marketing-how-we-use-contextual-signals-break-through-mobile-users/>, Aug., 2017. Accessed on Jan. 2019.
- [4] M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "Pios: Detecting privacy leaks in ios applications," in *NDSS*, 2011, pp. 177–183.
- [5] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, p. 5, 2014.
- [6] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill, "Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem," in *NDSS*, 2018.
- [7] J. Ren, M. Lindorfer, D. Dubois, A. Rao, D. Choffnes, and N. Vallina-Rodriguez, "Bug fixes, improvements,... and privacy leaks—a longitudinal study of pii leaks across android app versions," in *NDSS*, 2018.
- [8] J. Crussell, R. Stevens, and H. Chen, "MAdFraud: Investigating Ad Fraud in Android Applications," in *Proc. of MobiSys*, 2014, pp. 123–134.
- [9] S. Son, D. Kim, and V. Shmatikov, "What Mobile Ads Know About Mobile Users," in *NDSS 2016*.
- [10] P. Vines, F. Roesner, and T. Kohno, "Exploring adint: Using ad targeting for surveillance on a budget-or-how alice can buy ads to track bob," in *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, 2017, pp. 153–164.
- [11] S. Demetriou, W. Merrill, W. Yang, A. Zhang, and C. A. Gunter, "Free for all! assessing user data exposure to advertising libraries on android," in *NDSS*, 2016.
- [12] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.
- [13] C. Zheng, S. Zhu, S. Dai, G. Gu, X. Gong, X. Han, and W. Zou, "Smartdroid: an automatic system for revealing ui-based trigger conditions in android applications," in *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, 2012, pp. 93–104.
- [14] M. Backes, S. Bugiel, and E. Derr, "Reliable third-party library detection in android and its security applications," in *Proc. of CCS*, 2016, pp. 356–367.
- [15] C. Gbiler, J. Crussell, J. Erickson, and H. Chen, "Androidleaks: automatically detecting potential privacy leaks in android applications on a large scale," in *International Conference on Trust and Trustworthy Computing*, 2012, pp. 291–307.
- [16] N. Fruchter, H. Miao, S. Stevenson, and R. Balebako, "Variations in tracking in relation to geographic location," *arXiv preprint arXiv:1506.04103*, 2015.
- [17] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes, "Recon: Revealing and controlling pii leaks in mobile network traffic," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, pp. 361–374.
- [18] "Smartphone apps crushing mobile web time," <https://www.emarketer.com/Article/Smartphone-Apps-Crushing-Mobile-Web-Time/1014498>, Accessed on Jan. 2019.
- [19] "mitmproxy suit," <https://mitmproxy.org/>, Accessed at Jan. 2019.
- [20] L. Gomez, I. Neamtiu, T. Azim, and T. Millstein, "Reran: Timing-and touch-sensitive record and replay for android," in *Proc. of ICSE*, 2013, pp. 72–81.
- [21] "Justtrustme," <https://github.com/Fuzion24/JustTrustMe>, Accessed at Jan. 2019.
- [22] "disconnectme disconnect tracking protection list," <https://github.com/disconnectme/disconnect-tracking-protection/blob/master/services.json/>, Accessed on Jan. 2019.
- [23] "webxray domain owners list," <https://goo.gl/zZRQfX>, Accessed on Jan. 2019.
- [24] "Android ad networks," <https://www.appbrain.com/stats/libraries/ad/>, Accessed at Jan. 2019.
- [25] M. Y. Wong and D. Lie, "Intellidroid: A targeted input generator for the dynamic analysis of android malware," in *Proc. of NDSS*, 2016.