University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Computer Science and Engineering: Theses, Dissertations, and Student Research

Computer Science and Engineering, Department of

Summer 2019

User Privacy Leakage in Location-based Mobile Ad Services

Qicheng Lin
University of Nebraska - Lincoln, lin.qicheng@yahoo.com

USER PRIVACY LEAKAGE IN LOCATION-BASED MOBILE AD SERVICES

by

Qicheng Lin

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Qiben Yan

Lincoln, Nebraska

August, 2019

USER PRIVACY LEAKAGE IN LOCATION-BASED MOBILE AD SERVICES

Qicheng Lin, M.S.

University of Nebraska, 2019

Adviser: Qiben Yan

The online advertising ecosystem leverages its massive data collection capability to learn the properties of users for targeted ad deliveries. Many Android app developers include ad libraries in their apps as a way of monetization. These ad libraries contain advertisements from the sell-side platforms, which collect an extensive set of sensitive information to provide more relevant advertisements for their customers. Existing efforts have investigated the increasingly pervasive private data collection of mobile ad networks over time. However, there lacks a measurement study to evaluate the scale of privacy leakage of ad networks across different geographical areas. In this work, we present a measurement study of the potential privacy leakage in mobile advertising services conducted across different locations. We develop an automated measurement system to intercept mobile traffic at different locations and perform data analysis to pinpoint data collection behaviors of ad networks at both the app-level and organization-level. With 1,100 popular apps running across 10 different locations, we perform extensive threat assessments for different ad networks. Meanwhile, we explore the ad-blockers behavior in the ecosystem of ad networks, and whether those ad-blockers are actually capturing the users private data in the meantime of blocking the ads.

We find that: the number of location-based ads tends to be positively related to the population density of locations, ad networks collect different types of data across different locations, and ad-blockers can block the private data leakage.

DEDICATION

To my parents Lili Fan and Faxing Lin, to my friends. Thank you all for the support you gave me.

ACKNOWLEDGMENTS

I would like to show special appreciation towards my advisor Dr. Qiben Yan, for his supervision and support towards the completion of this work. Meanwhile, I would like say thank members of THINK Lab for the valued assistance.

Table of Contents

Chapter 1

Introduction

Digital advertising market has changed dramatically since the invention of mobile devices. Based on the statistic of Statista [6], desktop ad spending remains roughly the same (about 30 billion U.S. dollars) from 2011 to 2019, while mobile ad spending grows from 1.57 million U.S. dollars in 2011 to 50.84 billion in 2017. The tremendous growth in mobile ad spending is mainly due to the increasing popularity of mobile devices and apps. Based on the Flurry Analytics data [8], an average U.S. consumer spends 5 hours a day on mobile apps, which allows ad networks to collect private information through the mobile traffic. In particular, the mobile advertising networks have quickly become the primary users of location data. High-accuracy location-based mobile advertising (LBMA) allows advertisers to launch a targeted advertising campaign with 5–20 times better response rates. A customer survey [9] shows a majority (51%) of U.S. mobile device users are in favor of location-based targeted ads due to their relevancy. However, despite its popularity, the privacy implications of LBMA require further scrutiny.

Existing works on mobile privacy have identified privacy leakage issues through mobile ad libraries using static analysis [13], dynamic analysis [14], and network traffic analysis [21]. A recent study has identified different privacy leakage behaviors across different app versions [22]. Yet, factors relevant to LBMA, i.e., the differences in private information collection behaviors of mobile ad networks across different apps, organizations, and locations, have not been well investigated.

The primary challenge to identify geographical patterns of mobile privacy leakages resides in the astronomical number of geolocation involved. To collect the mobile traffic, the researcher must physically go to each location and collect mobile traffic. Naive approaches that spoof GPS signals can be easily detected by cross-referencing the GPS coordinates with the devices' network profiles, and users' regular activities.

To combat this, we develop a sophisticated method to reduce the location sample size and bypass location verification. Since mobile tracking is evolved from web tracking, the advertisers engaging in web tracking also be likely adopt mobile tracking. The relationship of mobile in-app tracking and web tracking from the same ad networks is useful for understanding the ad networks' private data collection capabilities. We, therefore, identify the hot zones and cold zones for mobile privacy study using the physical locations of the advertisers whose websites contain online tracking contents. To bypass location verification, we consistently spoof the GPS coordinates, network profile, and user activities, so they appear coherent in the eyes of ad networks.

Our method also allows us to understand how mobile ad networks aggregate information across multiple apps and the role of ad blockers play in this system. With the lack of tracking cookies in mobile apps, ad networks incorporate ad libraries in mobile apps to track users' private information. Ad libraries are embedded in multiple apps with different granted permissions. By linking the permission profiles of different active apps at different locations to the same ad network, we may understand how the ad network fuse user's information for targeted advertising. With some many users use ad blockers on their phone or tablet, we may raise the concern the ad blockers' capability of gathering users' data.

There are several challenges in the study of mobile ads ecosystem: There are a large number of ad networks in the market and each of them may have different targeting mechanisms and privacy policy, several ad networks may belong to the same organization, in order to fully understand the severity of privacy leakage, we need to associate these ad networks to their main organization, ad networks are using different ways to identify user's location, we need to take this into consideration while faking our device's location, there are some popular apps that implement SSL pinning which we could not intercept their traffic at all, older versions of Android has some technique called SSL unpinning but sometimes those popular apps require a newer versions of Android to run. We are trying to address these challenges in our measurement study.

Based on our extensive measurement with real-world apps, we discover that: 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 more 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 some particular types of private information; 4) most ad networks can infer users' precise location even if they do not collect the fine-grained GPS coordinates; 5) ad blockers can block a lot of private information leaked to third parties without gathering the data.

Our contributions are summarized below.

- We design an efficient privacy leakage measurement system to conduct a location-based study by automatically adjusting the locations, conducting traffic collection, and performing traffic analysis. We develop domain classification mechanisms to 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. It is alarming that the ad network organizations can collect a comprehensive set of users' private information by aggregating data from multiple apps.

- We conduct extensive measurement to expose ad networks' information collection behaviors across different locations. Our results suggest that ad networks have different private information collection behaviors at different locations; specifically, they collect different types of private information across different cities. Location leakage by ad networks is particularly disconcerting, as most ad networks can either collect or infer precise locations. The ad blockers can block a lot of ads and they stopped significant personal information leakage towards the advertisers or third parties also. According to our result, the ad blockers do not have the feasible access to the data they blocked.

Chapter 2

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 [11]. Consequently, the new SSPs and DSPs have incorporated security features into their products [24]. Researchers have raised concerns of malicious advertisers recently [24, 25], who can obscure the apps' background to hide malicious activities. In response, most DSPs are rapidly improving their screening process to filter out malicious ads and require a minimum number of targeted audiences to prevent individual targeting [25].

Demetriou et al. [12] presents the first measurement system to bring to light 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 [22, 20, 21]. 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 [13], malware classification [17], and user activity analysis [27]. The changes across different versions of ad libraries [10] 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 [14]. The location leakage through location-based services (LBS) has been analyzed [19, 16, 15]. In this paper, we analyze apps across different cities in the 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 [26]. Customizable tools like Android Studio's Monkey[1] and Appium[2] allow researchers to provide a customized simulation of app interactions.

[1] https://developer.android.com/studio/test/monkey

[2] http://appium.io/docs/en/about-appium/intro/

Chapter 3

Background

3.1 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.3.1. 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 (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 applica-

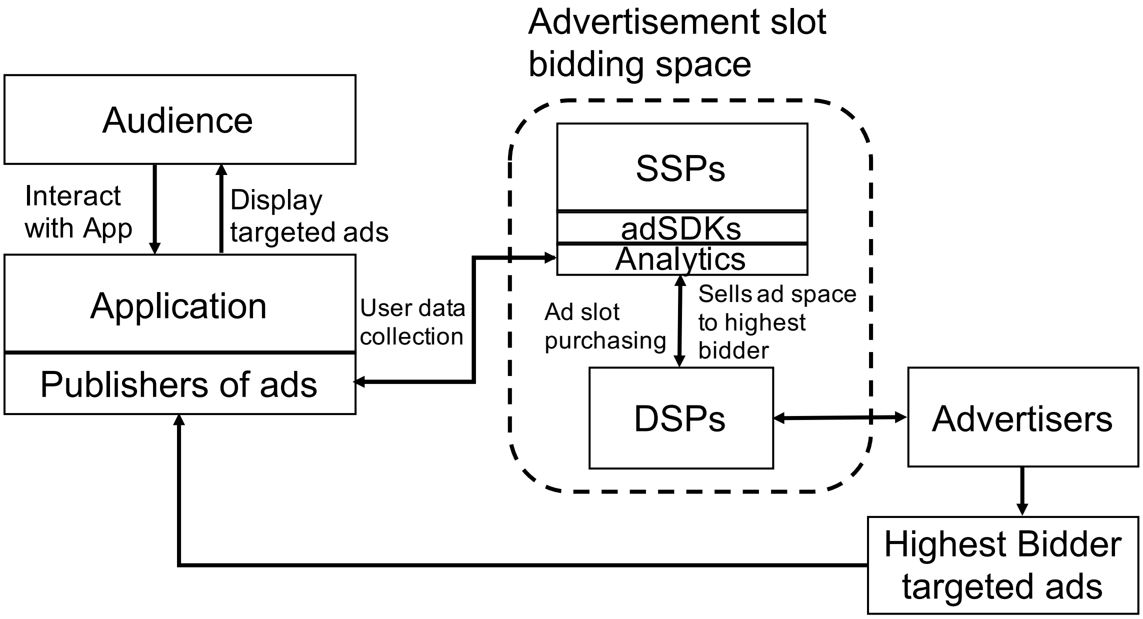


Figure 3.1: Mobile advertisement ecosystem.

tion stimulus, which collects private data protected by the 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 advertising identifiers (MAIDs), locations, network profiles, device types, etc.

3.2 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 determine if the cross-application ad libraries can correlate the multiple instances of

TABLE 3.1: 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 3.2: 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

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 [21, 22, 23], 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, listed in Table 3.1.

3.3 Threat Model

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

Threat from a organization with multiple domains. Popular ad networks usually contain multiple third-party services to aggregate more comprehensive private information from different domains. The ad networks are able to collect users' private information across multiple apps. Therefore, 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 collecting private information to serve more ads.

Threat from network eavesdroppers. Networks eavesdroppers may get private information by listening to the network communications. Some of the private information may be leaked in plaintext via HTTP. For instance, we know that the ad blockers can help us to block a lot of annoying ads from the advertisers, and they play the role as middle men in this system, but no one can guarantee their innocuous besides the claim they made for themselves. In our study, we try to evaluate the severity of such privacy leakage and understand what information an eavesdropper can obtain. Meanwhile, we want to know whether those ad blockers behave the same way as they claimed.

Chapter 4

Location-Based Measurement Platform

Our measurement platform mainly consists of two components: location-based traffic measurement and traffic analysis. Fig. 4.1 shows the overall structure of our measurement platform to collect and analyze traffic to identify privacy leakage of ad networks across different locations.

In-app advertising and mobile web advertising both have their advantages and limitations in the eyes of advertisers. According to eMarketer [4], mobile apps account for nearly 86% of time spent using smartphones. But a few top apps dominate the app usage. In fact, based on a recent study [3], the top 5 apps takes nearly 85% of the total app usage time, which means advertisers may need to spend most of their budget on a handful dominating companies. On the other hand, 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, suggesting that mobile web advertising may reach a more diverse set of audiences. 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.

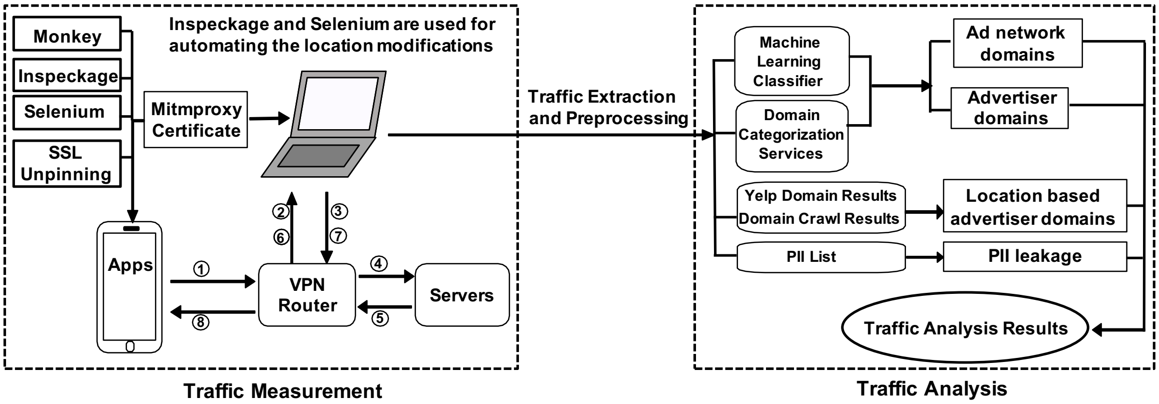


Figure 4.1: **The proposed platform** consists of traffic measurement and traffic analysis. Mitmproxy has been set up as a transparent HTTPS proxy. ① The mobile client initiates a connection to the server. ② The router redirects the connection to mitmproxy. ③ mitmproxy dynamically generates certificates 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.

4.1 Traffic Measurement

Our traffic measurement consists both real mobile devices and emulators, a wireless router, and a workstation. Mitmproxy [2] 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 [23, 22], 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. 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.

The location-based study of this paper 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 to fake the IP addresses, which are configured to be consistent with 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 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.

To study the behavior of ad blockers, we selected ten popular andorid ad blcoker apps to test the capability for each of them. We first run the ad blocker app, and then for each app, we let it run for ten minutes with auto event injection by Monkey. We use the same methodology we proposed above to capture all the traffics. Since for this small part, location changing is not one of our concerns, so we do not apply the IP changing and fake GPS to this part.

4.2 Traffic Analysis

For analysis, we focus on identifying and extracting information from network traffic related to the ad networks. We propose a domain classification mechanism to extract the third-party domains, ad network domains, advertise domains, and location-based ad domains. Meanwhile, we catogorize different types of ad blockers.

4.2.1 Identify third-party domains

Domains can be classified as first-party domains and third-party domains, and the owner of first-party domains are the app's owner. To differentiate between the first-party and third-party domains, 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 our traffic, we count each unique domain. To avoid bias, we only count once if the domain appears multiple times in one app. We identify all the third-party domains, whose number of appearance is higher than the threshold. Because 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.

4.2.2 Identify ad network domains

We first generate a list of ad network domains using the publicly available information and two domain organization mapping list on GitHub [7, 5]. 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 network domain and get descriptive information of each domain. 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 set using 2,000 non-ad network domains from Alexa, and another 2,000 randomly picked ad network domains from EasyList for training and testing. In the end, the domain classification accuracy is around 70%.

To improve the classification performance, we also use three classification engines, i.e., VirusTotal, McAfee, and OpenDNS, to generate the domain classification result to be our ground truth. For each unique domain we find in the traffic, we query the classification engines for every domain which provide us the related information such as category, subdomains, and the feedback of Whois lookup for the queried domains. If one of the engines considers the domain as an ad domain, we will add it to our list. We evaluate the performance using our ground-truth data with labels, and the domain classification accuracy can achieve 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, but also many small ad networks which have insignificant market shares.

4.2.3 Identify advertiser domains

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 - VirusTotal, McAfee, and OpenDNS. 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.

4.2.4 Identify location-based ad domains

After classifying the advertisers' domains, we still need to identify whether the advertiser's related ads are location-based ads. To verify it, we have to identify the relationship between the advertiser domain and the specific locations of the served ads. Here, we consider the local ads as the location-based ads, which can be confirmed by Yelp. Simply relying on Yelp's query results may not be sufficient in identifying the local businesses. To differentiate the local business with all the advertisers, we crawl all the advertiser domains and check if the front page of each domain contains the city name or not. By combining the yelp local business list and web crawling result, we can identify the location-based ad domains.

4.2.5 Identify ad blocker types

There are majority of two types of ad blockers, the first one is based on the blacklist and whitelist, and the second one is based on the VPN to filter all the ad related traffics. The first type generally need to require the root access of the phone in order to change the system settings based on their predefined blacklist and whitelist. Therefore, all the traffics fall into the blacklist will be blocked by the system which are those ads. Vise versa, the second type do not need the root access and they normally have their own VPN built on their server, all the outgoing and incoming traffics will go through their VPN before reach the phone. In another words, the traffics that lands on the phone have been filtered by the VPN server without any ads. To differentiate those ad blockers, we used the detector built in the android system. After we launch the app, we go to the system setting to check whether there is a new VPN running. To further identify whether those VPN ad blockers' capability of capturing our PII data on their server side. we need to identitfy the exact VPN protocols those ad blockers used, and we use wireshark as our tool to verify this. First, we make sure their is no VPN running on the phone, and we start to dump all the traffics by launching the VPN ad blockers. Then we use wireshark to check all those dumped traffics. There is a column in wireshark called ”protocol”, and we first go through all the traffics manually to see whether wireshark can help us to identify the VPN protocols. Unfortunately, wireshark only provides us the protocols like TCP, HTTP, or TLS, etc., which are not the ones we are looking for. In the end, we find out there are regular ports used for each of the VPN protocols, and we start to filter traffics through regular ports, which help us to exclude some protocols.

4.2.6 Identify PII leaks

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. For PII leakage study, we first extract the HTTP/HTTPs request URL, response URL, and request/response contents. By integrating the domain organization mapping lists mentioned above [7, 5], 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.

Algorithm 1 PII Leakage Identification Algorithm

INPUT: Predefined PII list (according to Table 3.1), Domain organization mapping list.
OUTPUT: PII leakage of each app

```
for each App do
    for each location do
        Extract Gets and Posts URLs from captured traffic flows
        Extract key-value pairs from the URLs
        Match the key-value pairs with hashed PII values in PII list
        if find a match then
            Log the key-value pair as a PII leakage for the app
            Extract domains associated with the key-value pair
            Match domains to the domain organization mapping list
            if find a match then
                Log the key-value pair as a PII leakage for the matched organization
            else
                Log the key-value pair as a PII leakage for “Others” organization
            end if
        end if
    end for
end for
Return PII leakage results
```

Chapter 5

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.

5.1 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 5.6. 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.

5.2 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.

Fig. 5.2 shows the number of third-party domains embedded in the landing page of web ads sorted by domain names, and the number of third-party domains in the traffic of mobile apps.

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. Please refer to [18] for detailed result.

Our experimental results in Figure 5.1 show the distribution of HTTP field values in the collected network traffic from the ad libraries, and the two rightmost columns show that most advertisers specify their landing URLs in HTTP (more than 95%) rather than HTTPS (less than 5%). The reason is that many third-party contents embedded within the landing pages are loaded over HTTP, which can cause mixed-content errors if the original sites are upgraded to use HTTPS. The low adoption rate of HTTPS in web ad deliveries is likely to continue since third parties use HTTP by default to better serve HTTP referrer headers and advertise the sources of the

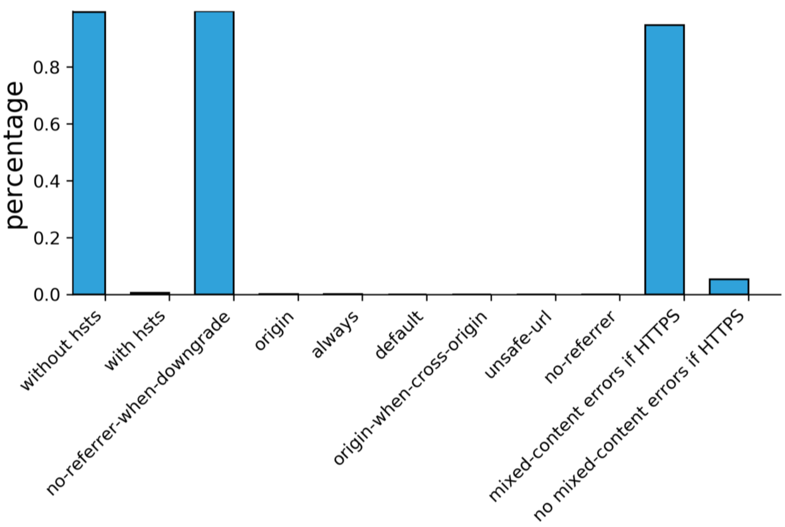


Figure 5.1: HTTP field distribution among all the ad libraries redirected traffic.

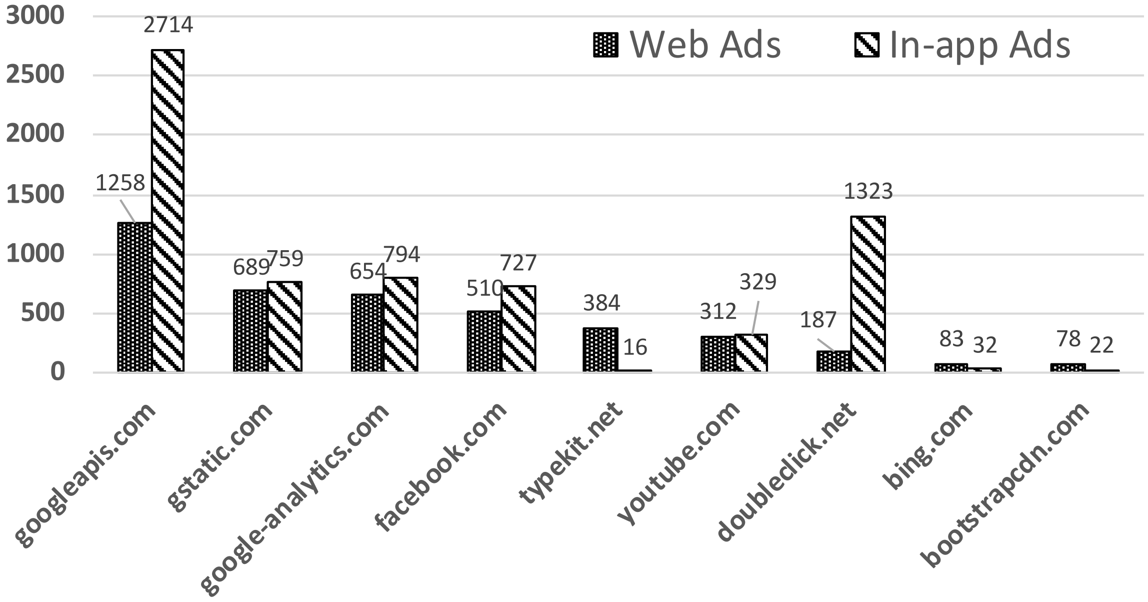
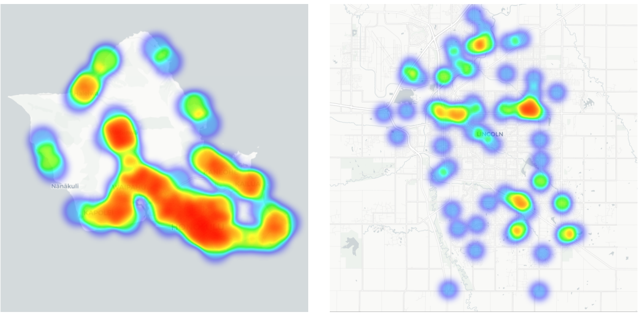


Figure 5.2: Third party domains for mobile web ads and in-app ads.

Finding 2: mobile web ads have a significantly lower adoption rate of 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 5.1, we can see that HTTP traffic dominates in the 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 5.1: 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%



(a) Oahu (b) Lincoln

Figure 5.3: Heatmap of location requests in landing URLs on the island of Oahu and Lincoln.

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 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. 5.3 shows the heatmap of location requests

TABLE 5.2: 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

(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.

5.3 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 [1] 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 of which can be found in [18]. 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 5.2 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

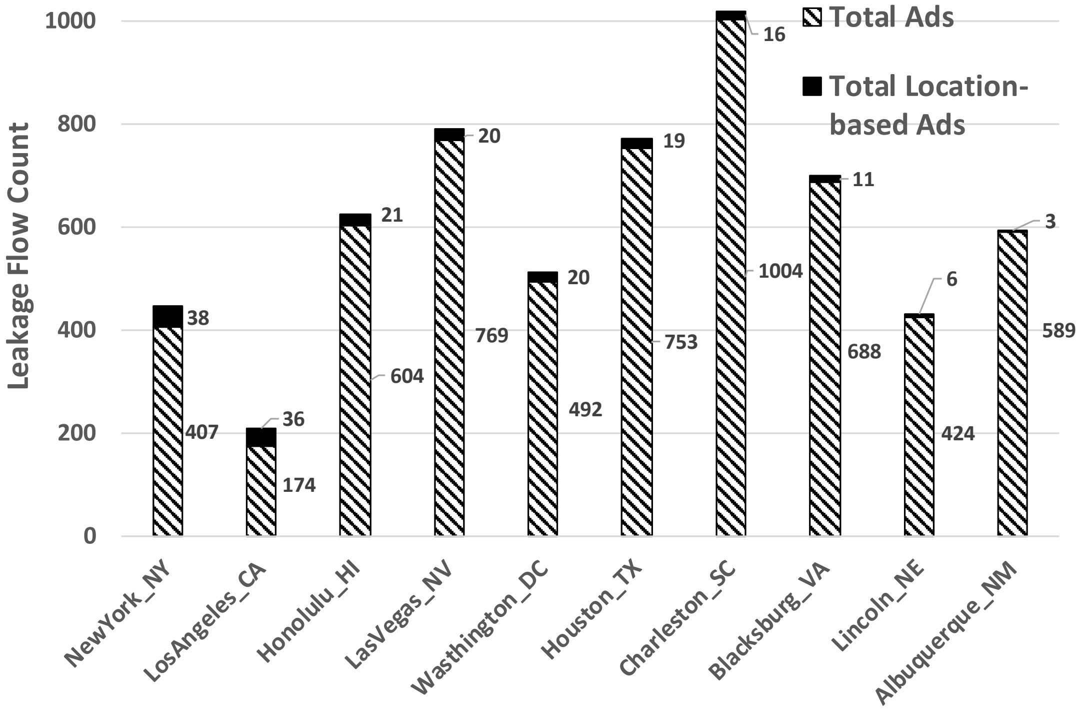


Figure 5.4: Number of total ads and location based ads in each city (Dataset_2).

to 496 (or 247) different organizations. Using the domain classification methods in Section 4.2, we can identify 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. 5.4 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 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.

TABLE 5.3: Top 10 ad organization list of PII leakage severity in all locations sort 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

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. The detailed result can be found in [18]. 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 5.3 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 5.4: 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 5.5: 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

5.4 Location-based Private Data Collection of Ad Networks

Ad networks extensively collect users' location information. Table 5.4 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

TABLE 5.6: 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 5.7: 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

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 5.5 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 5.6 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. 5.5 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 5.7, 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 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.

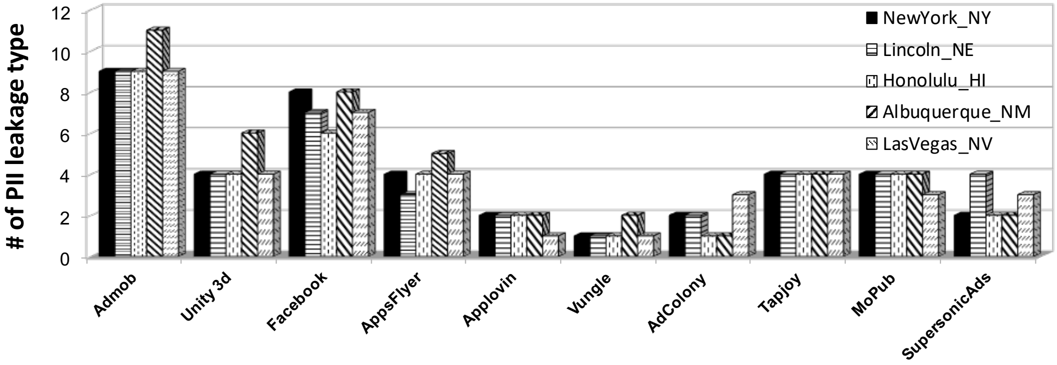


Figure 5.5: The leaking PII types of top 10 ad networks across different locations.

Finding 7: most ad networks collect fine-grained GPS location data. Table 5.8 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.

TABLE 5.8: 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
1rx.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

5.5 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 3.2, 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. The details of these selected apps can be found in [18]. 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: More diverse group of mobile ad networks show up 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 5.9. 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 5.9: 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

TABLE 5.10: PII leakage w/o ad blockers

Ad blcokers	Average # of GPS	Average # of IP	Average # of Zip code	Average # of Android ID
without adblockers	118	56	210	74
with adblockers	47	32	75	49

5.6 The Data Collection of Ad Blockers

Finding 9: Ad Blockers can efficiently block the PII leakage without gathering the user's PII. There are different kinds of ad blockers as we mentioned in the section 4.2.5, so here we listed all the attributes that related to the ad blockers we explored in this paper as shown in Table 5.11. As we can tell, half of the ad blockers we explored used VPN and half of them used blacklist approach. Meanwhile, we are based on the regular ports that are used by different VPN protocols to filter the traffic which we listed in Table 5.12. In the traffic, we didn't find any UDP packages, and we found the port 443 has been largely used, so we could narrow down the VPN types to OpenVPN or SSTP, which both of are identified to be secure VPN protocols. In another words, based on our experiments, those ad blockers do not have feasible access to the users' traffic. The last objective for us is to explore the capabilities of ad blockers in the process of leaking users PII to those third-parties. As shown in Table 5.10, majority of ad blockers can block a lot of ad traffics and PII leakage at the same time.

TABLE 5.11: Ad blcoker attributes

Ad blocker Name	Root needed	Blacklist	VPN	VPN Types	Mobile App or Browser
AdGuard	No	No	Yes	OpenVPN or SSTP	App
AdClear	No	No	Yes	OpenVPN or SSTP	App and Browser
AdAway	Yes	Yes	No	/	App and Browser
Brave	No	Yes	No	/	Browser
Block This	No	No	Yes	OpenVPN or SSTP	App and Browser
Ad Lock	No	No	Yes	OpenVPN or SSTP	App and Browser
Ad Blocker Plus	No	Yes	No	/	App and Browser

TABLE 5.12: VPN Type and Regular ports

VPN Types	TCP regular ports	UDP regular ports	Security level
OpenVPN	502, 501, 443, 110, 80	1194, 1197, 1198, 8080, 9201, 53	very secure
L2TP	/	500, 1701, 4500	not secure
PPTP	1723	/	not secure
IKEv2	/	500, 1701, 4500	not secure
SST	433	/	very secure

Chapter 6

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 mechanism 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 [26] 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 4.2, 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.

VPN Protocol Identification. In this paper, we used the regular ports to find out the potencial VPN protocols used by those ad blockers. However, we only explored limited number of ad blocker apps, and our approach to find out the protocol types can be enhanced. For instance, different VPN types have different handshakes when we connect to the VPN server, and one idea is to find out the patterns of handshakes in the traffic for different VPN protocols, so we could use that as our ground truth to further evaluate our result.

Chapter 7

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. The ad blockers can effectively block a lot of traffics that related to PII leakage, and they do not have the feasible access to the user's data.

Bibliography

[1] Android ad networks. https://www.appbrain.com/stats/libraries/ad/, Accessed at Jan. 2019.

[2] mitmproxy suit. https://mitmproxy.org/, Accessed at Jan. 2019.

[3] Nearly 85 percent of smartphone app time concentrated in top five apps. https://marketingland.com/nearly-85-percent-smartphone-app-time-concentrated-top-five-apps-report-191624/, Accessed at Jan. 2019.

[4] Smartphone apps crushing mobile web time. https://www.emarketer.com/Article/Smartphone-Apps-Crushing-Mobile-Web-Time/1014498, Accessed at Jan. 2019.

[5] webxray domain owners list. https://goo.gl/zZRQfX, Accessed on Jan. 2019.

[6] 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.

[7] disconnectme disconnect tracking protection list. https://github.com/disconnectme/disconnect-tracking-protection/blob/master/services.json/, Accessed on Jan. 2019.

[8] 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.

[9] Which mobile location ads are the most accurate? https://www.emarketer.com/Article/Which-Mobile-Location-Ads-Most-Accurate/1011226, Sep., 2014. Accessed on Jan. 2019.

[10] Michael Backes, Sven Bugiel, and Erik Derr. Reliable third-party library detection in android and its security applications. In *Proc. of CCS*, pages 356–367, 2016.

[11] Jonathan Crussell, Ryan Stevens, and Hao Chen. MAdFraud: Investigating Ad Fraud in Android Applications. In *Proc. of MobiSys*, pages 123–134, 2014.

[12] Soteris Demetriou, Whitney Merrill, Wei Yang, Aston Zhang, and Carl A Gunter. Free for all! assessing user data exposure to advertising libraries on android. In *NDSS*, 2016.

[13] Manuel Egele, Christopher Kruegel, Engin Kirda, and Giovanni Vigna. Pios: Detecting privacy leaks in ios applications. In *NDSS*, pages 177–183, 2011.

[14] William Enck, Peter Gilbert, Seungyeop Han, Vasant Tendulkar, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)*, 32(2):5, 2014.

[15] Nathaniel Fruchter, Hsin Miao, Scott Stevenson, and Rebecca Balebako. Variations in tracking in relation to geographic location. *arXiv preprint arXiv:1506.04103*, 2015.

[16] Clint Gibler, Jonathan Crussell, Jeremy Erickson, and Hao Chen. Androidleaks: automatically detecting potential privacy leaks in android applications on a large scale. In *International Conference on Trust and Trustworthy Computing*, pages 291–307, 2012.

[17] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*, 2016.

[18] Boyang Hu, Qicheng Lin, Yao Zheng, Qiben Yan, Matthew Troglia, and Qingyang Wang. Characterizing location-based mobile tracking in mobile ad networks. https://www.dropbox.com/s/8c0gvs2wir5s55k/cns-2019-mobile-technical-report.pdf?dl=0, Jan. 2019.

[19] Siyuan Ma, Zhushou Tang, Qiuyu Xiao, Jiafa Liu, Tran Triet Duong, Xiaodong Lin, and Haojin Zhu. Detecting gps information leakage in android applications. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 826–831, 2013.

[20] Wei Meng, Ren Ding, Simon P Chung, Steven Han, and Wenke Lee. The price of free: Privacy leakage in personalized mobile in-apps ads. In *NDSS*, 2016.

[21] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *NDSS*, 2018.

[22] Jingjing Ren, Martina Lindorfer, Daniel Dubois, Ashwin Rao, David Choffnes, and Narseo Vallina-Rodriguez. Bug fixes, improvements,... and privacy leaks–a longitudinal study of pii leaks across android app versions. In *NDSS*, 2018.

[23] Jingjing Ren, Ashwin Rao, Martina Lindorfer, Arnaud Legout, and David 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*, pages 361–374, 2016.

[24] Sooel Son, Daehyeok Kim, and Vitaly Shmatikov. What Mobile Ads Know About Mobile Users. In *NDSS 2016*.

[25] Paul Vines, Franziska Roesner, and Tadayoshi 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*, pages 153–164, 2017.

[26] Michelle Y. Wong and David Lie. Intellidroid: A targeted input generator for the dynamic analysis of android malware. In *Proc. of NDSS*, 2016.

[27] Cong Zheng, Shixiong Zhu, Shuaifu Dai, Guofei Gu, Xiaorui Gong, Xinhui Han, and Wei 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*, pages 93–104, 2012.