Sebastian Zimmeck*, Peter Story*, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel Reidenberg, N. Cameron Russell, and Norman Sadeh*

MAPS: Scaling Privacy Compliance Analysis to a Million Apps

Abstract: The app economy is largely reliant on data collection as its primary revenue model. To comply with legal requirements, app developers are often obligated to notify users of their privacy practices in privacy policies. However, prior research has suggested that many developers are not accurately disclosing their apps' privacy practices. Evaluating discrepancies between apps' code and privacy policies enables the identification of potential compliance issues. In this study, we introduce the Mobile App Privacy System (MAPS) for conducting an extensive privacy census of Android apps. We designed a pipeline for retrieving and analyzing large app populations based on code analysis and machine learning techniques. In its first application, we conduct a privacy evaluation for a set of 1,035,853 Android apps from the Google Play Store. We find broad evidence of potential non-compliance. Many apps do not have a privacy policy to begin with. Policies that do exist are often silent on the practices performed by apps. For example, 12.1% of apps have at least one location-related potential compliance issue. We hope that our extensive analysis will motivate app stores, government regulators, and app developers to more effectively review apps for potential compliance issues.

DOI 10.2478/popets-2019-0037

***Corresponding Author: Sebastian Zimmeck:** Department of Mathematics and Computer Science, Wesleyan University, E-mail: szimmeck@wesleyan.edu. The first two authors contributed equally to this study. Previously, Sebastian Zimmeck was a postdoctoral associate at Carnegie Mellon's School of Computer Science.

***Corresponding Author: Peter Story:** School of Computer Science, Carnegie Mellon University, E-mail: pstory@andrew.cmu.edu.

Daniel Smullen, Abhilasha Ravichander, Ziqi Wang: School of Computer Science, Carnegie Mellon University.

Joel Reidenberg, N. Cameron Russell: School of Law, Fordham University.

***Corresponding Author: Norman Sadeh:** School of Computer Science, Carnegie Mellon University, E-mail: sadeh@cs.cmu.edu.

1 Introduction

Privacy legislation around the world sets requirements for the disclosure of privacy practices. Such laws include the Children's Online Privacy Protection Act (COPPA) and the California Online Privacy Protection Act (CalOPPA) in the US and the General Data Protection Regulation (GDPR) in the EU. These and other laws are creating stricter and more extensive obligations for app developers to make their privacy practices transparent by means of privacy policies. Despite their well-known shortcomings — they take a long time to read [33] and are often difficult to understand [14] — privacy policies remain the *de iure* standard for notifying Internet users of applicable privacy practices.

When it comes to the dissemination of these policies for apps, app stores are situated at a critical juncture. In 2012, six major app stores signed an agreement with the California Attorney General in which they obligated themselves to urge developers to adopt privacy policies and to provide functionality allowing developers to link to their policies [6]. Consequently, on the Play Store, Google is requiring Android developers to disclose how their apps collect, use, and share user data [20]. However, we are not aware of Google or any other Android app store operator engaging in a systematic and comprehensive review of apps' privacy practices.

1.1 Research Questions

In order to advance the fundamental understanding of privacy practices and potential compliance issues in the Android ecosystem, we are introducing the Mobile App Privacy System (MAPS). Its design, implementation, and use cases are guided by the following research questions.

1. How many apps have privacy policies? (§ 5.1) Prior work has measured the prevalence of links to privacy policies [54], but it is also of significant interest whether privacy policies can actually be found by following these links.

2. Which privacy practices are developers describing in their privacy policies? (§ 5.2) Which practices are discussed the most? Are developers describing the practices that their apps are performing or are they making promises to users about what they will not do?
3. Of the practices performed by apps, which are described in privacy policies? (§ 5.3) If an app accesses users' information, but that practice is not described in a privacy policy, a compliance issue may be present. Why does this happen? Are the privacy practices of third parties described less often than those of first parties?
4. What characteristics of apps are associated with potential compliance issues? (§ 5.4) For example, do apps in certain Play Store categories have more potential compliance issues?

1.2 Key Contributions

In order to answer our research questions, we leverage MAPS. Our system compares apps' privacy practices to what their privacy policies state, and flags potential privacy requirements conflicts, which we characterize as potential compliance issues. Such issues arise when the claims made (or absent) in the privacy policy conflict with evidence that the app may be doing something else. Our system's evaluation of app behaviors is based on efficient and lightweight code analysis techniques. For the policy analysis, we leverage natural language processing and supervised machine learning models. The system is capable of performing large-scale scans: we use it to analyze over a million apps on the Google Play Store.

We believe that our system holds promise for app developers, app stores, privacy activists, and regulators alike. For example, regulators could focus their investigative efforts on those apps which our system has flagged as having potential compliance issues. Automatically identifying potential compliance issues could help app stores moderate their stores more effectively.

We make the following contributions:

1. APP-350 Corpus (§ 3). The machine learning classifiers of our system were trained based on a corpus of human-annotated app privacy policies. This corpus is publicly available for further research.[1] To our knowledge, it is the only corpus for app privacy policies and the largest corpus of annotated privacy policies overall.
2. MAPS (§ 4). Our system provides a scalable pipeline to systematically analyze potential compliance issues for sizable populations of apps. By evaluating apps' permissions, Android API usage, library inclusion, privacy policies, and numerous types of metadata, it enables a comprehensive evaluation of privacy trends.
3. Google Play Store Privacy Analysis (§ 5). Based on our system's analysis, we present an extensive privacy survey of 1,035,853 free Android apps on the Google Play Store. Our analysis finds broad evidence of potential compliance issues. Particularly, many apps appear to not sufficiently disclose third party practices concerning identifiers and locations. We notified regulators of some of our findings (§ 6), and performed a pilot study with a large European electronic device manufacturer (§ 7).

This work has been conducted as part of the Usable Privacy Policy Project [46] and the Personalized Privacy Assistant Project.[2]

2 Related Work

Our work builds on prior work in large-scale privacy studies (§ 2.1). We also employ code analysis (§ 2.2) and natural language processing (§ 2.3) techniques.

2.1 Privacy Surveys

Our study is extending prior work on identifying privacy practices in mobile apps and comparing them against disclosures in privacy policies. As lawmakers introduce new privacy legislation, privacy policies remain the primary means for disclosing and describing a service's privacy practices. Such policies, together with statutory and other laws, provide the ground truth for objectively measuring privacy compliance. Consequently, privacy policies have been used in the past for privacy grading metrics [60, 66].

We are unaware of prior work analyzing potential compliance issues across entire app stores. However, several studies have performed privacy analyses of smaller

[1] The dataset is available at https://data.usableprivacy.org.

[2] The Personalized Privacy Assistant Project, https://privacyassistant.org.

numbers of apps. Zimmeck et al. [67] previously analyzed potential compliance issues for a set of 17,991 apps. Wang et al. [61] analyzed 80 health and finance apps for compliance with their privacy policies, detecting 20 "strong" and 10 "weak" violations. Using TaintDroid, Enck et al. [13] performed an automated dynamic privacy analysis of 30 popular Android apps.

Other studies have focused on data sharing with third parties. In a longitudinal study Ren et al. [43] observed 512 Android apps over eight years of version history and concluded that the increased number of third party domains receiving data lead to higher privacy risk over time. Because third party libraries and their host apps have access to the same Android app permissions, it is often difficult to discern who is processing what data. Thus, end users are often forced to accept privacy-relevant third party behavior if they want to make use of the apps' functionality [24]. Although their study was focused only on websites and not mobile apps, Libert [26] audited the disclosure of third party data collection practices on 200,000 privacy policies for websites. He found that the names of third parties are usually not explicitly disclosed in website privacy policies. In line with these observations, we examine the state of potential privacy non-compliance with regard to third party practices in particular.

2.2 Android App Analysis

The privacy practices of interest in our study are the subject of studies in both the privacy and security communities. Viennot et al. [59] found that more than half of the apps they examined contained a third party ad library. Some existing approaches [51, 64], however, are not capable of distinguishing between first and third party practices, limiting the utility of analysis under existing legal frameworks.[3] Razaghpanah et al. [41] made use of Lumen (previously Haystack) [40] to develop methods for detecting previously unknown third party libraries and uncovering relationships between parties. Liu et al. found that native code use in ad libraries makes up only about 1% [28]. It has negligible impact on app populations as a whole [1], and is therefore omitted from our study.

Many studies have explored app behavior by measuring data leakage. Examining apps with WebViews, Mutchler et al. [34] found that 28% of apps leak data through overridden URLs or have similar vulnerabilities. The aim of our research questions is distinct from this area of research, as it is our objective to analyze apps' privacy practices — we are not exploring patterns of exploiting security vulnerabilities. We are aiming to identify disclosed use of permissions and APIs [36] instead of detecting maliciously hidden information flows and service invocations. In this context Tuncay et al. [58] found that the coexistence of certain app permissions can lead to unintentional loss of privacy. Our focus also differs from the goal of tools like DroidSafe [22] and ISA [25].

Using dynamic analysis a previous study [45] revealed that many Android apps track users via persistent device identifiers, a practice Google prohibits for advertising purposes [21]. Our method of triangulating app behavior is based on static analysis. One of the most popular static analysis tools is FlowDroid [2]. Our approach is based on FlowDroid's notion that the execution of particular APIs is indicative of certain privacy practices, as shown in the Appendix, Table T2. However, FlowDroid's runtime performance and reported error rate of 102/477 app analyses given a 30-minute time-out window [51] is not suitable for our purposes. We opted to develop a lightweight analysis approach that scales well and is sufficiently robust, similar to the approach used in [27]. Alternative techniques apply machine learning [31, 35, 38, 44], black box differential analysis [9], or traffic signatures [7].

2.3 Privacy Policy Analysis

To answer our privacy policy-related research questions our work leverages natural language processing and machine learning techniques, which stands in contrast to Watanabe et al. [62], whose approach is based on keywords. Using a naive Bayes classifier, Zimmeck and Bellovin [66] built a browser extension for identifying privacy practices in policy text. Tesfay et al. [55], also using machine learning, identified various GDPR provisions in policies. Many approaches to analyzing policy text are based on supervised machine learning techniques that require expert annotated training and test datasets. Previously, a corpus of annotated website policies was released by Wilson et al. [63] and used by Harkous et al. [23]. We contribute to this effort by releasing a privacy policy corpus specifically for apps (§ 3).

[3] For example, CalOPPA requires operators of online services to disclose in their privacy policies whether "other parties" are collecting personally identifiable information on their services. Cal. Bus. & Prof. Code §22575(b)(6).

It is a challenge to identify which natural language text documents are indeed privacy policies as, for example, privacy policy links in the Play Store may be broken or redirect to non-policy documents. Also, privacy policies may be combined with terms of services. Story et al. [54] studied metadata concerning apps on the Play Store and found that many apps lack privacy policy links on their Play Store pages altogether. Earlier work focused on the structure of privacy policies in more general domains outside of Android apps. Fei Liu et al. [39] and Frederick Liu et al. [29] addressed the problem of identifying policy sections relating to the same topic. Sathyendra et al. [47] classified advertising opt outs and other privacy-related options on websites. Cranor et al. [10] evaluated financial institutions' privacy notices. Bowers et al. [5] studied privacy policies of mobile money services, and Ermakova et al. [14] analyzed the readability of policies of healthcare websites. Zhuang et al.'s work [65] aimed to help university researchers by automating enforcement of privacy policies of Institutional Review Boards.

3 The APP-350 Corpus

Our system analyzes privacy policy text based on supervised machine learning classifiers. In order to train the models and test their performance we created an app privacy policy corpus — the APP-350 corpus — which is available for further research.[4] Legal experts identified and annotated policy text relating to *privacy practices*, or short *practices*, such as an app's access of GPS location information by a first or third party. The nature of a practice may be general (e.g., "We access your location information") or specific (e.g., "We access your GPS location information").

A policy may describe the performance of a practice (e.g., "We access your location information"), make the promise that a practice is not performed (e.g., "We do not access your location information."), may contain statements on both performance and non-performance, or may not mention a practice at all. As all policies were comprehensively annotated with performance and non-performance labels, it can be assumed that all unannotated portions of policy text do not describe any of the practices and so can be used as training, validation, and test data to detect the absence of statements on respective practices.

The legal experts annotated a total of 350 policies. We selected these policies from the most popular apps on the Google Play Store. We annotated the policies linked from the Play Store pages of all apps with more than 50 million installs ($n = 247$). In addition, we annotated the policies of randomly selected apps with more than 5 million installs ($n = 103$). All 350 policies were consistently annotated by one of the authors, who is a lawyer with experience in data privacy law. As the reliability of our classifiers depends on the quality of annotations, we hired two law students unrelated to our project. As recommended [4], the students double-annotated 10% of the corpus. We paid the students $14/hour and asked each to independently annotate a set of 35 policies that we randomly selected from the full corpus of 350 policies. For evaluating the reliability of the annotations we measured agreement among the three annotators based on Krippendorff's α, which indicates agreement to be good above 0.8, fair between 0.8 and 0.67, and doubtful below 0.67 [30].

With a mean of Krippendorff's $\alpha = 0.78$ the agreement levels generally exceed previously reported results [67]. Detailed results are shown in the Appendix, Table T1. Generally, inter-annotator agreement levels in the privacy policy domain tend to be relatively low due to the vagueness and ambiguity of policy language. For example, prior work reported $\alpha = 0.48$ for identifying statements on location sharing [67]. The lowest agreement level for a practice we included in MAPS was for SIM Serial access by third parties, with Krippendorff's $\alpha = 0.49$.[5] However, low levels of agreement do not necessarily present a problem; classifiers can achieve good performance despite being trained on data with low inter-annotator agreement, so long as the disagreement looks like random noise [42], as was the case for the practices we used in MAPS.

Promises to not perform a certain practice are fairly uncommon in privacy policies. Due to the rarity of negative annotation labels in our corpus, we enriched 142 randomly selected policies from our training and validation sets with synthetic data; we added sentences with

[4] The dataset is available at https://data.usableprivacy.org.

[5] In preliminary tests we also considered city, ZIP code, postal address, username, password, ad ID, address book, Bluetooth, IP address (identifier and location), age, and gender practices. However, we ultimately decided against further pursuing those as we had insufficient data, unreliable annotations, or difficulty identifying a corresponding API for the app analysis.

negative annotation labels by manually changing policy text from a positive modality to a negative modality. We apply the most common forms of negation [56] with the same aggregate probability distribution as they appeared in the rest of our corpus. Using this approach, the synthetic data matches the statistical character of the data collected through our annotation task.

4 Scaling the Privacy Analysis

We designed MAPS to efficiently and reliably analyze large numbers of apps. Based on a pipeline of distributed tasks (§ 4.1), our system achieves sufficient runtime performance (§ 4.2). MAPS is comprised of separate modules for the analysis of policies (§ 4.3), apps (§ 4.4), and potential compliance issues (§ 4.5).

4.1 Pipeline of Distributed Tasks

Our system begins its analysis by recursively crawling apps' Play Store pages by following links to similar apps [54]. Each newly discovered app's metadata, such as the app's Play Store categories and privacy policy URL, if any, are stored in a database.

Apps are downloaded using a fork of an unofficial Python Google Play API,[6] which we configured with Google account credentials associated with a Google Pixel phone running Android 7.1.2 to ensure compatibility with a large number of apps. As more than 90% of apps in the Play Store are free [54] and costs would be prohibitive, we did not analyze paid apps. Each app is decompiled into Smali bytecode with Apktool[7] as part of the app analysis (§ 4.4). In order to identify links to privacy policies inside apps, our system performs a search for relevant URLs in the Smali bytecode and the apps' `strings.xml` resource files. If a URL contains either the words "privacy," "policy," or "legal," it will be identified as a policy URL.

The system uses headless Firefox browsers to download privacy policies using the URLs found on Play Store pages and in decompiled apps. Using a real browser instead of a single HTTP request has the advantage of being able to capture dynamically loaded JavaScript content. Our system handles HTML and PDF policies. In a test with 110 links our browser properly retrieved 105 documents. However, as some potential policy URLs lead to pages that are not policies (e.g., homepages), our system uses a logistic regression classifier [49] to identify (English-language) policies. Based on a test set of 100 retrieved documents with 65 positive instances, the classifier achieves 99.0% accuracy and a 99.2% F1 score. The test set was labeled independently by two authors, who agreed on 100% of the labels. As some policy URLs are for privacy landing pages (e.g., lists of policies for different countries), our system performs a limited crawl using the policy classifier to identify privacy policies. After the policy download, the system performs the privacy policy analysis to determine the practices described therein (§ 4.3). The system's final step is the comparison of an app's policy and app analyses to identify potential compliance issues (§ 4.5).

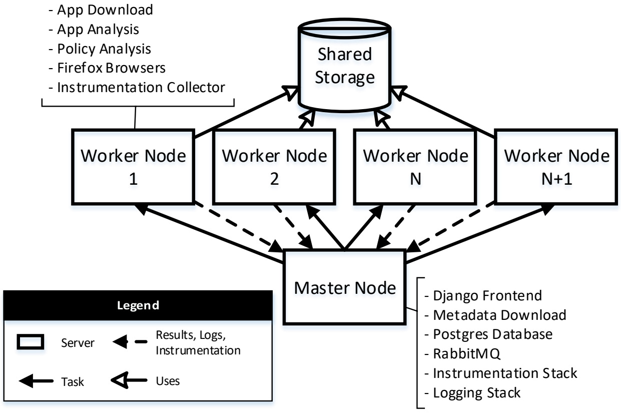


Fig. 1. A simplified depiction of our distributed system consisting of a master node controlling multiple worker nodes.

Our system runs on a cluster of distributed computers. The system's long-term storage is hosted at one of our institutions, the master node on Amazon EC2,[8] and the worker nodes on XSEDE's TACC Jetstream cluster [57]. Figure 1 shows a general overview. Docker[9] is used to encapsulate dependencies, with each task corresponding to at least one container. We use Docker Swarm[10] to distribute our system's containers across worker nodes. Our system uses Celery[11] with

[6] Python Google Play API, https://github.com/NoMore201/googleplay-api, accessed: June 19, 2019.

[7] Apktool, https://ibotpeaches.github.io/Apktool/, accessed: June 19, 2019.

[8] Amazon EC2, https://aws.amazon.com/ec2/, accessed: June 19, 2019.

[9] Docker, https://www.docker.com/, accessed: June 19, 2019.

[10] Swarm mode overview, https://docs.docker.com/engine/swarm/, accessed: June 19, 2019.

[11] Celery—Distributed Task Queue, http://celery.readthedocs.io/en/latest/, accessed: June 19, 2019.

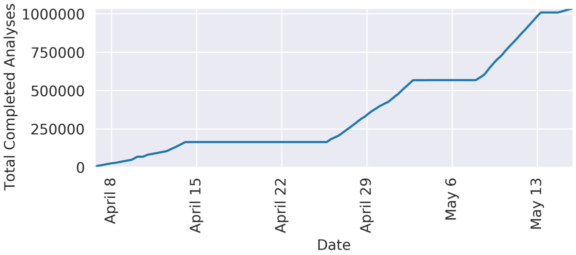


Fig. 2. The progression of app analyses during our Play Store sweep, which ran for 512 hours at an average rate of 2,023 apps/hour.

RabbitMQ[12] to distribute tasks and collect analysis results. In order to monitor system health and troubleshoot problems, we aggregate and search system logs using Graylog.[13] We further make use of Telegraf,[14] InfluxDB,[15] and Grafana[16] to visualize time-series instrumentation data, such as CPU usage, load averages, and the rates of task completion.

4.2 Hardware and Runtime Performance

We performed our Play Store analysis from April 6 to May 15, 2018 as depicted in Figure 2. Our fastest rate of 4,529 apps/hour was achieved on May 8 with 10 complete and several small worker nodes. Each complete worker node runs all of the services in Figure 1 and has 6 virtual CPUs, 16 GB of memory, and 60 GB of disk space. We also use small worker nodes with 2 virtual CPUs, 4 GB of memory, and 20 GB of disk space. Small worker nodes only run the app download task. One small worker node also hosts the shared storage to store downloaded apps prior to analysis. Our master node is an Amazon EC2 t2.2xlarge instance with 8 virtual CPUs, 32 GB of memory, and 200 GB of gp2 SSD storage.

Of the 1,049,790 retrieved apps, 1,035,853 (98.67%) were analyzed successfully. Of the apps which were not analyzed successfully, 1.03% failed to download, 0.21% failed in the app analysis, 0.08% failed in the policy analysis, and 0.01% failed during our re-analysis.[17] The 1,039,003 apps we downloaded occupy approximately 13TB of storage. Compared to an earlier Play Store crawl from April 24 through June 22, 2013 that lead to 5.3TB of data for about 960,000 apps [59], the average app size more than doubled over the last five years, from about 5MB to about 13MB. This increase could be the result of more app code, more third party library code, or larger resources (e.g., to support higher-resolution screens).

4.3 Privacy Policy Analysis

We characterize the detection of privacy practice descriptions in privacy policies as a classification task. The goal of the task is to assign annotation labels to policy segments, that is, structurally related parts of policy text that loosely correspond to paragraphs [29, 63]. We decompose the classification task into three subtasks: classifying (1) data types (e.g., `Location`), (2) parties (i.e., `1stParty` or `3rdParty`), and (3) modalities (i.e., whether a practice is explicitly described as being performed or not performed).[18] For example, the `Performed Location Cell Tower 3rdParty` classification will be assigned to a segment if the `Location Cell Tower`, `3rdParty`, and `Performed` classifiers all return a positive result for the segment. The decomposition of the classification task allows for an economic use of annotated data. We randomly divided the APP-350 corpus into training ($n = 188$), validation ($n = 62$), and held-out test ($n = 100$) sets, the latter of which we only used to calculate classifier performance.

As classifier performance depends on adequate preprocessing of policy text as well as domain-specific feature engineering, we normalize whitespace and punctuation, remove non-ASCII characters, and lowercase all policy text. Because stemming did not lead to performance improvements, we are omitting it. In order to run our classifiers on the most relevant set of features, we use an optional preprocessing step of sentence filtering. Based on a grid search, in cases where it improves

[12] RabbitMQ, https://www.rabbitmq.com, accessed: June 19, 2019.

[13] Graylog, https://www.graylog.org, accessed: June 19, 2019.

[14] Telegraf, https://github.com/influxdata/telegraf, accessed: June 19, 2019.

[15] InfluxData, https://www.influxdata.com, accessed: June 19, 2019.

[16] Grafana, https://grafana.com, accessed: June 19, 2019.

[17] After completing the Play Store analysis we noticed a bug in our app analysis code. As a result, we re-ran the app analyses and re-calculated all statistics. 135 additional analyses failed yielding a final total of 1,035,853 successfully analyzed apps.

[18] Note that the `Single Sign On` and `Single Sign On: Facebook` practices do not use a party classifier, as all data is exchanged between the app developer as first party and the SSO provider as third party.

classifier performance, we remove a segment's sentences from further processing if they do not contain keywords related to the classifier in question [67]. For example, the `Location` classifier is not trained on sentences which only describe cookies. We identified relevant keywords based on a manual review of segments from our training and validation data and added additional synonyms. Sentence filtering improved the performance of about half our classifiers.

Prior to training, we generate vector representations of the segments. Specifically, we take the union of a TF-IDF vector and a vector of manually crafted features. Our TF-IDF vector is created using the TfidfVectorizer [48] configured with English stopwords (`stop_words='english'`), unigrams and bigrams (`ngram_range=(1, 2)`), and binary term counts (`binary=True`). This configuration is similar to what was used in prior work [29]. Our vector of manually crafted features consists of Boolean values indicating the presence or absence of indicative strings we observed in our training and validation data. For example, we include the string `not collect`, because we assumed it would be a strong indicator of the negative modality.

For all but four classifiers, we used scikit-learn's SVC implementation [50] and trained with a linear kernel (`kernel='linear'`), balanced class weights (`class_weight='balanced'`), and a grid search with five-fold cross-validation over the penalty parameter (`C=[0.1, 1, 10]`) and gamma parameter (`gamma=[0.001, 0.01, 0.1]`). For four data types (`Identifier`, `Identifier IMSI`, `Identifier SIM Serial`, and `Identifier SSID BSSID`), we created keyword based rule classifiers due to the limited amount of data and their superior performance.

Table 1 shows the performance of the classifiers. For example, we say a policy describes a first party practice, that is, there is a + Support instance for a first party practice, if the data type, first party, and positive modality classifiers return a positive result for at least one policy segment. If that is not the case for any of the segments, a - Support instance exists. Since our definition of a potential compliance issue does not depend on the negative modality classifier, we do not include it in Table 1. A segment can contain multiple first and/or third party practices. As potential compliance issues are dependent on practices being *not* described in policies [67], negative predictive value, specificity, and negative F1 are particularly meaningful performance metrics. With negative F1 scores ranging from 78% to 100%, 23 of the classification tasks achieved higher negative F1 scores than the closest comparable previous classifiers and 3 performed equally [67]. Our results reveal that generally + Support is lower for third party practices, that is, third party practices are often not as extensively described in privacy policies as first party practices. For additional details about our classifiers' performance, including error and ablation analyses, please consult our related study [53].

Policy Classification	NPV	Specificity	Neg. F1	+/- Support
Contact 1stParty	92%	96%	94%	30/70
Contact 3rdParty	95%	96%	95%	8/92
Contact Email Address 1stParty	78%	90%	84%	80/20
Contact Email Address 3rdParty	91%	83%	87%	13/87
Contact Phone Number 1stParty	93%	93%	93%	54/46
Contact Phone Number 3rdParty	97%	93%	95%	5/95
Identifier 1stParty	93%	68%	**78%**	20/80
Identifier 3rdParty	97%	76%	85%	8/92
Identifier Cookie 1stParty	100%	92%	96%	63/37
Identifier Cookie 3rdParty	94%	92%	93%	52/48
Identifier Device ID 1stParty	86%	96%	91%	54/46
Identifier Device ID 3rdParty	97%	95%	96%	21/79
Identifier IMEI 1stParty	99%	99%	99%	17/83
Identifier IMEI 3rdParty	99%	100%	99%	4/96
Identifier IMSI 1stParty	100%	100%	**100%**	3/97
Identifier IMSI 3rdParty	99%	100%	99%	1/99
Identifier MAC 1stParty	95%	98%	96%	19/81
Identifier MAC 3rdParty	99%	96%	97%	6/94
Identifier Mobile Carrier 1stParty	90%	100%	95%	21/79
Identifier Mobile Carrier 3rdParty	98%	97%	97%	3/97
Identifier SIM Serial 1stParty	100%	97%	98%	8/92
Identifier SIM Serial 3rdParty	100%	99%	99%	1/99
Identifier SSID BSSID 1stParty	99%	100%	99%	5/95
Identifier SSID BSSID 3rdParty	100%	99%	99%	0/100
Location 1stParty	92%	81%	86%	58/42
Location 3rdParty	96%	83%	89%	23/77
Location Cell Tower 1stParty	98%	94%	96%	14/86
Location Cell Tower 3rdParty	98%	95%	96%	4/96
Location GPS 1stParty	99%	94%	96%	29/71
Location GPS 3rdParty	99%	94%	96%	6/94
Location WiFi 1stParty	99%	86%	92%	12/88
Location WiFi 3rdParty	100%	95%	97%	2/98
Single Sign On	89%	90%	90%	37/63
Single Sign On: Facebook	95%	84%	89%	32/68

Table 1. Classifier performance for determining whether a policy states that a practice is performed in our policy test set ($n = 100$). Negative predictive value (NPV) and Specificity are precision and recall for negative instances, respectively. Negative F1 (Neg. F1) is the F1 for negative instances. In the Support column, + is the number of ground truth positive instances (i.e., a policy truly describes a practice being performed) and - is the number of ground truth negative instances (i.e., a policy truly does not describe a practice being performed).

4.4 Android App Analysis

Since MAPS is designed to detect potential compliance issues at app store-wide scale, we needed to employ lightweight, reliable techniques for analyzing apps' practices. After decompiling apps into Smali,[19] our system operates on four app resources: Android APIs, strings, permissions, and class structure. We assume a threat model which considers data as compromised from the moment a privacy-sensitive API appears to be called [36]. Table T2 in the Appendix lists the APIs our system searches for in the Smali bytecode. As some location and identifier API behavior is dependent on a string parameter (e.g., the `GPS_PROVIDER` string), our system also performs a call graph analysis to trace relevant strings. Since an API call will fail if an associated Android permission is not granted, the system checks that the APIs' permissions are included in the app's `AndroidManifest.xml` file. First and third party classes are distinguished based on Java's reverse domain name notational convention [37]. In particular, a `.smali` file's package name is compared to the app's Play Store package name: if (1) both top and second level domain match, (2) the file is not part of any package, or (3) the `.smali` file's package appears to be obfuscated (e.g., `a/b.smali`), the API call is considered a first party call. Otherwise, it is categorized as a third party call. Note that our app analysis does not involve taint tracking, which appears infeasible to perform at app store-wide scale [51]. Detecting apps' usage of Facebook's Single Sign On functionality is dependent on whether it retrieves an access token via Facebook's `getCurrentAccessToken()` API and whether it contains a Facebook app ID string.

To verify our system's analysis we performed a manual dynamic evaluation of 100 apps, each of which is associated with one of our test policies. If a developer provided more than one app under the same policy, we selected one randomly. We ran each app on a rooted Android Moto X phone with an Xposed[20] module we wrote. 17 apps could not be analyzed due to forced automatic app updates, apps' refusal to run on a rooted phone, or failures in API logging. For some observed calls we were not able to detect whether it came from a first or third party. In those instances we searched the decompiled app code in order to determine if all call sites originated in either first or third party classes, in which case we declare a first or third party practice, respectively. Otherwise, the result is dismissed as unknown. Our approach for distinguishing first and third parties, according to our three step process described above, is subject to an error rate of 7% for third parties who should have been categorized as first parties. There were no first parties that should have been categorized as third parties.

App Practice	Precision	Recall	F1	+/-/? Support
Contact Email Address 1stParty	84%	100%	91%	26/55/19
Contact Email Address 3rdParty	47%	89%	62%	9/72/19
Contact Phone Number 1stParty	92%	100%	96%	11/72/17
Contact Phone Number 3rdParty	50%	100%	67%	4/79/17
Identifier Cookie 1stParty	63%	100%	77%	**12**/59/29
Identifier Cookie 3rdParty	82%	100%	90%	**50**/21/29
Identifier Device ID 1stParty	87%	98%	92%	40/39/21
Identifier Device ID 3rdParty	97%	100%	99%	75/4/21
Identifier IMEI 1stParty	88%	100%	94%	22/59/19
Identifier IMEI 3rdParty	75%	96%	84%	28/53/19
Identifier IMSI 1stParty	44%	100%	62%	4/77/19
Identifier IMSI 3rdParty	73%	100%	85%	11/70/19
Identifier MAC 1stParty	74%	100%	85%	14/67/19
Identifier MAC 3rdParty	58%	100%	74%	25/56/19
Identifier Mobile Carrier 1stParty	85%	97%	91%	35/46/19
Identifier Mobile Carrier 3rdParty	93%	100%	97%	71/11/18
Identifier SIM Serial 1stParty	50%	100%	67%	3/80/17
Identifier SIM Serial 3rdParty	50%	100%	67%	8/75/17
Identifier SSID BSSID 1stParty	80%	100%	89%	12/71/17
Identifier SSID BSSID 3rdParty	52%	100%	68%	16/67/17
Location Cell Tower 1stParty	100%	88%	93%	8/70/22
Location Cell Tower 3rdParty	91%	95%	93%	22/56/22
Location GPS 1stParty	83%	100%	91%	5/72/23
Location GPS 3rdParty	79%	88%	83%	17/60/23
Location WiFi 1stParty	100%	86%	92%	7/71/22
Location WiFi 3rdParty	81%	89%	85%	19/59/22
Single Sign On: Facebook	81%	57%	67%	30/53/17

Table 2. Performance for determining the practices in our app test set ($n = 100$). In the Support column, + and - are the respective numbers of ground truth positive and negative instances and ? is the number of instances for which the manual analysis failed or the ground truth is unknown.

Table 2 shows the app analysis performance as compared to the ground truth of our manual evaluation. If our system's static analysis flagged a practice, we counted a true positive if we manually observed the practice and a false positive if we did not (e.g., due to unreachable code). If the static analysis did not flag a practice, we counted a true negative if we did not manually observe the practice and a false negative if we did manually observe the practice (e.g., due to code obfuscation). In contrast to our policy results (§ 4.3), it can be observed that many practices are performed by third parties more often than by first parties. For example, we observe 50 + Support instances for `Identifier Cookie 3rdParty`, but only 12 for `Identifier Cookie 1stParty`. This result is consistent with earlier findings of nearly every ad library leaking phone data and, if available, location data [17]. It further suggests that many apps may not be compliant with Google's prohibition of using device identifiers for purposes of ad

[19] The decompilation is based on Apktool (version 2.3.1.), https://ibotpeaches.github.io/Apktool/, accessed: June 19, 2019.

[20] Xposed Installer, http://repo.xposed.info/module/de.robv.android.xposed.installer, accessed: June 19, 2019.

Potential Compliance Issue	Precision	Recall	F1	+/-/? Support
Contact Email Address 1stParty	75%	75%	75%	4/77/19
Contact Email Address 3rdParty	38%	71%	50%	7/74/19
Contact Phone Number 1stParty	100%	100%	100%	1/82/17
Contact Phone Number 3rdParty	29%	67%	40%	3/80/17
Identifier Cookie 1stParty	50%	100%	67%	1/70/29
Identifier Cookie 3rdParty	83%	87%	85%	23/48/29
Identifier Device ID 1stParty	70%	88%	78%	16/63/21
Identifier Device ID 3rdParty	96%	86%	91%	**58**/21/21
Identifier IMEI 1stParty	79%	65%	71%	17/64/19
Identifier IMEI 3rdParty	76%	85%	80%	26/55/19
Identifier IMSI 1stParty	33%	67%	44%	3/78/19
Identifier IMSI 3rdParty	69%	82%	75%	11/70/19
Identifier MAC 1stParty	83%	91%	87%	11/70/19
Identifier MAC 3rdParty	58%	78%	67%	23/58/19
Identifier Mobile Carrier 1stParty	78%	70%	74%	20/61/19
Identifier Mobile Carrier 3rdParty	92%	75%	83%	**64**/18/18
Identifier SIM Serial 1stParty	50%	50%	50%	2/81/17
Identifier SIM Serial 3rdParty	50%	88%	64%	8/75/17
Identifier SSID BSSID 1stParty	83%	56%	67%	9/74/17
Identifier SSID BSSID 3rdParty	53%	62%	57%	16/67/17
Location Cell Tower 1stParty	100%	100%	100%	2/76/22
Location Cell Tower 3rdParty	79%	73%	76%	**15**/63/22
Location GPS 1stParty	N/A	N/A	N/A	0/77/23
Location GPS 3rdParty	70%	70%	70%	**10**/67/23
Location WiFi 1stParty	50%	100%	67%	1/77/22
Location WiFi 3rdParty	75%	75%	75%	**12**/66/22
Single Sign On: Facebook	56%	45%	50%	11/72/17

Table 3. Performance for detecting potential compliance issues on our test set of app/policy pairs ($n = 100$). In the Support column, + and - are the respective numbers of ground truth positive and negative instances of potential compliance issues, and ? is the number of instances where missing ground truth data from our app analyses makes it unclear whether such issues exist. N/A is shown where the metrics are undefined, or where a lack of positive ground truth instances would always make the metric zero.

tracking [21]. App developers integrate a diverse set of third party libraries in their apps. Beyond ad networks and analytics services, many developers leverage developer frameworks, which may be less privacy-invasive. However, in a random sample of 50 instances comprising libraries of five popular developer frameworks, we found that in 68% (34/50) there was at least one ad network or analytics service performing the same practice as a developer framework.

4.5 Compliance Analysis

We define a *potential compliance issue*, or short *potential issue*, to mean that an app is performing a privacy practice (e.g., a first party is accessing GPS location data) while its associated privacy policies do not disclose it either generally (e.g., “Our app accesses your *location* data.”) or specifically (e.g., “Our app accesses your *GPS* data.”). Table 3 shows our system’s identification of potential compliance issues and its performance. For the 26 practices for which positive ground truth instances were present, we observe a mean F1 score of 71%. Many potential compliance issues relate to the access of identifier information by third parties, most notably, the access of mobile carrier (64 + Support instances) and device identifier (58 + Support instances) information. However, the three third party location practices Cell Tower, GPS, and WiFi account for 15, 10, and 12 respective + Support instances as well. Notably, all third party practices exhibit a higher number of potential compliance issues than their first party counterparts.

We conducted a statistical analysis of our techniques in order to determine the effect of our error rates on the large scale analysis results in the subsequent section. Our statistical analysis, as described in the Appendix and depicted there in Figure F1, strongly suggests that our techniques provide a sound basis on which we can reliably build our large-scale analysis.

5 What Is the State of Privacy in the Google Play Store?

Our large-scale analysis of free apps in the Google Play Store provides us with a rich dataset for evaluating the state of privacy in a substantial part of the Android ecosystem. In particular, we examine how many apps have privacy policies (§ 5.1) and what practices are discussed in them (§ 5.2). We also analyze how prevalent potential compliance issues are (§ 5.3) and what app characteristics are associated with their occurrence (§ 5.4).

5.1 How Many Apps Have Policies?

Oftentimes, apps are required to have a privacy policy, for example, if they collect personally identifiable information from California or Delaware residents.[21] However, as shown in Figure 3, our analysis reveals that only 50.5% of apps have links to privacy policies on their Play Store pages. In addition, our app analysis found links to policies in the code of 4.4% of apps. However, 4% of these apps also have a policy link on their Play Store page, so the retrieval of policy links from inside the app only marginally increased policy coverage by 0.4%.

[21] Cal. Bus. & Prof. Code §22575(a) and Del. Code Tit. 6 §1205C(a).

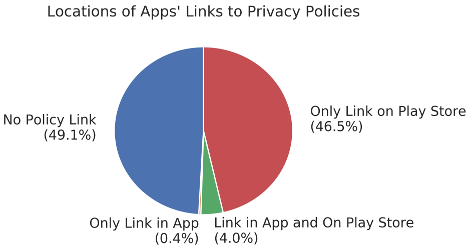


Fig. 3. Locations where apps' privacy policy links, if any, can be found.

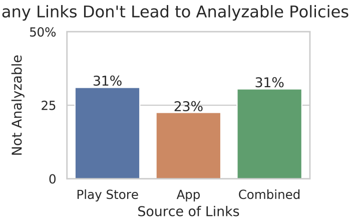


Fig. 4. The percent of apps with privacy policy links which do not lead to analyzable policies (i.e., English-language policies).

Overall, 49.1% of apps do not have privacy policy links, despite the fact that our analysis of apps' practices shows that 88.6% of apps perform at least one practice. These findings suggest that Google does not comprehensively enforce its requirement for developers to disclose their apps' privacy practices, especially as developers "must ... [p]ost a privacy policy in both the designated field in the Play Console and within the app itself" to the extent their app handles "sensitive user data includ[ing] ... personally identifiable information" [20].

Of those apps with Play Store and/or in-app policy links, our system identified that 31% are still missing analyzable privacy policies, as shown in Figure 4. Many of these apps' links redirected to invalid domains, 404 pages, non-English policy pages, and pages with non-privacy related content. This finding is especially noteworthy as our system does not require the links to directly point to a privacy policy. For links found on the Play Store, MAPS performs a limited crawl at the destination page in order to identify policies which are linked indirectly (§ 4.1).

Also, as we performed our analysis for the US version of the Play Store, policy links leading to non-English privacy policies highlight the challenge of enforcing privacy-related rules and laws in a global community of app developers. Consequently, we believe it would be promising to extend our analysis in future work to build classifiers for other languages or to test our classifiers' performance on automatically translated non-English webpages and privacy policies.

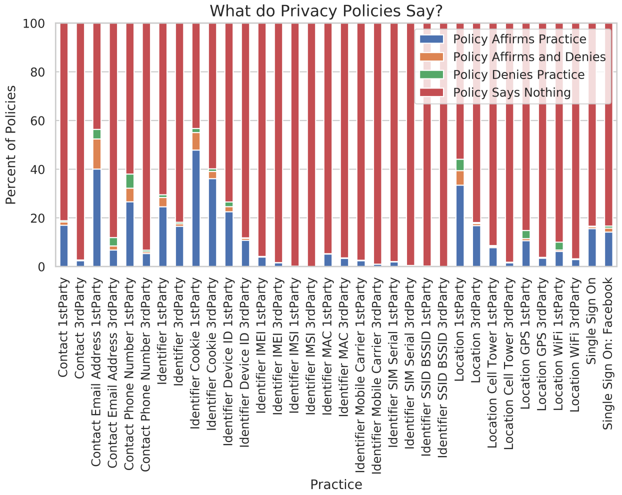


Fig. 5. Third party practices are discussed less frequently than first party practices, which is precarious given that the former are usually more opaque from a user's perspective. Note that a policy both affirming and denying a practice at the same time does not necessarily represent a contradiction, but could also indicate its discussion in different contexts (e.g., "We disclose your phone number to advertisers, but not to data brokers.").

5.2 Which Practices are Described in Policies?

Figure 5 depicts the occurrence of policy statements relating to the practices we examine. It can be observed that most practices are described only infrequently; that is, a policy does not mention it at least once. Further, the statements that are present typically affirm that a practice is occurring. This finding reveals that users seem to be given little assurance of potentially objectionable practices not being performed (e.g., disclosing users' phone numbers to third parties). Silence about privacy practices in privacy policies is problematic because there are no clear statutory default rules of what the privacy relationship between a user and a service should be, in the absence of explicit statements in the policy [32].

The Federal Trade Commission (FTC) — the main arbiter of privacy notice and choice in the US — is moving towards requiring "complete" or "meaningful" privacy notices [52]. For example, in United States v. Path [16], the defendant disclosed that its app collects "certain information [...], such as your Internet Protocol (IP) address, your operating system, the browser type,

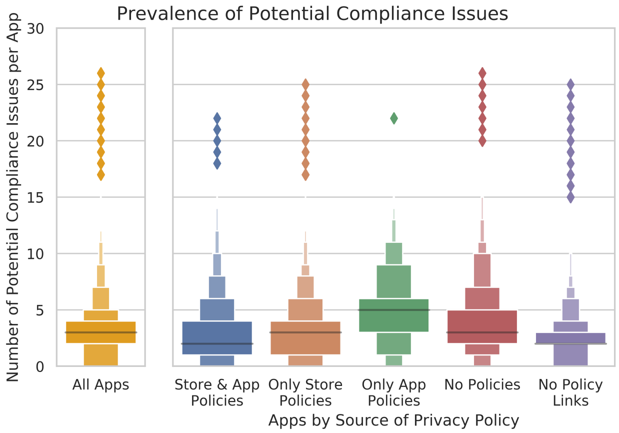


Fig. 6. The distribution of the number of potential issues broken down based on whether and where the apps' policies were found. The leftmost plot covers all apps — with and without policies and independent of policy location. Store & App Policies were found by following both links from the app's Play Store page and from the app code. No Policies means that despite the presence of links on the Play Store or in the app, no policies were found by following those links. No Policy Links means that there was neither a policy link on the app's Play Store page nor in its code.

the address of a referring site and your activity on our site." However, the FTC did not find this language sufficient to cover the allegedly wrongfully collected contact data and determined Path's privacy policy to be incomplete [52]. Thus, the relatively sparse discussion of privacy practices — which continues to exist in many apps' policies despite the wide adoption of ad libraries [59] — suggests broad evidence of non-compliance, which we will explore in detail in the subsequent sections.

5.3 How Many Apps Have Potential Compliance Issues?

Our system allows for identification of potential compliance issues, that is, instances in which an app performs a practice but does not have a policy which affirms that the practice is performed. Note that when our system finds multiple privacy policies for a given app, it pools the practice descriptions across all policies, which makes the analysis results more conservative. Overall, we measure a mean of 2.89 potential compliance issues per app and a median of 3. However, as Figure 6 shows, there is a significant amount of variation in the number of potential issues depending on whether an app has a policy and where its link is located.

The Prevalence of Potential Compliance Issues Differs Depending on Policy Link Location

It is striking that the number of potential issues for apps which only have policy links in their code ("Only App Policies") is higher than for all other categories, even higher than for apps with no policy links at all. However, as shown in Figure 3, the number of such apps is relatively small (0.4%). While apps that do not process personally identifiable information may not need a policy, it appears that many apps without one ("No Policies" and "No Policy Links") are in fact performing pertinent practices and, consequently, exhibit potential issues. It should be noted that some of the apps for which our system did not find a policy may have a policy in a language other than English. These apps would be included in the "No Policies" category of Figure 6, which may explain why the median and mean numbers of potential compliance issues are higher for the "No Policies" category than for the "No Policy Links" category (median: 3 vs 2; mean: 3.93 vs 2.50). To err on the side of caution, avoiding detection of potential issues in cases where apps have policies in languages other than English, we are omitting the apps in the "No Policies" category ($n = 160,695$) in all subsequent figures and statistics, which makes the analysis results more conservative. However, in principle, it is not the burden of the user to translate policies from another language that is not commonly spoken in the country in which goods or services are offered.[22]

Performance of a Practice is Correlated with the Occurrence of a Potential Issue

Figure 7 demonstrates that in most cases the performance of a practice is strongly correlated with the occurrence of a potential issue: if a practice is performed, then there is a good chance a potential issue exists as well. This result suggests a broad level of potential noncompliance. Identifier-related potential issues are the most common. Three different types of identifiers make up most potential issues: cookies, device IDs, and mobile carriers. In particular, the use of device IDs may constitute a misuse for purposes of ad tracking [21]. We also observed elevated numbers of location-related potential compliance issues. 15.3% of apps perform at least

[22] See generally GDPR, Recital 23.

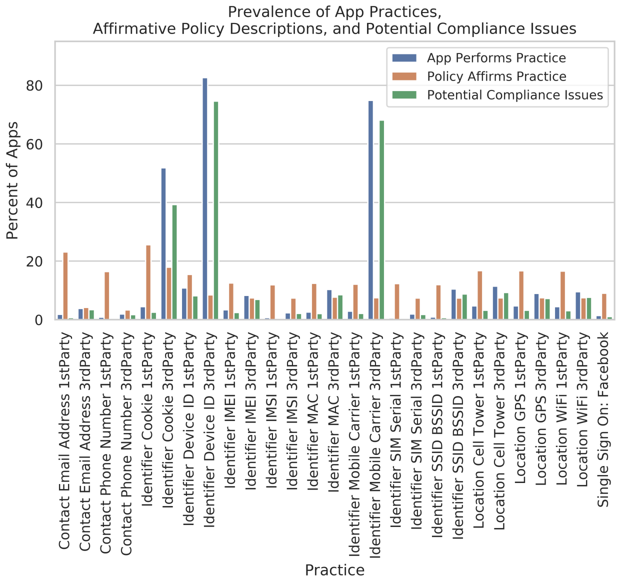


Fig. 7. The percent of apps performing the examined practices alongside the percent of apps whose policies affirmatively describe the practices and the percent of apps with potential issues. The percents of apps describing the practices are calculated based on the practice being described either specifically or generally (§ 4.5). Note that some policies describe practices which are not performed by their apps, which is why there can be more descriptions than practices performed.

one location-related practice, and 12.1% of apps have at least one location-related potential issue.

Third Party Practices Are More Opaque Than First Party Practices

For all data types, third party practices are more common than first party practices, and so are third party-related potential issues. Figure 8 shows the most frequently occurring advertising and analytics-related third party packages in our dataset. One reason for the prevalence of potential issues for third party practices could be that app developers are unaware of the functionality of the libraries they integrate. However, the developer documentation for each of the ten third party services in Figure 8 obligate developers to obtain user consent for the data processing in connection with the third party code integration. For example, the Google Analytics Terms of Service [19] obligate the developer to explicitly disclose the library integration in their privacy policy. Furthermore, various laws require app developers to disclose third party practices of their apps in their

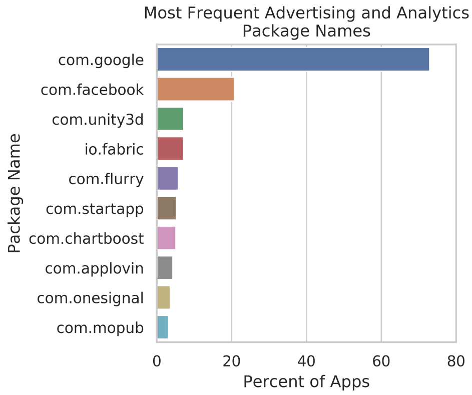


Fig. 8. Packages are only counted if they performed at least one of the practices we studied. Percents are calculated based on a random sample of 10,000 apps. Note that com.google combines AdMob, Google Analytics, and other Google services.

own privacy policy.[23] Under those laws, it is not sufficient to include a statement that the user should consult the third party's privacy policies.

App developers might benefit from assistance in their task of disclosing third party data processing. Our results suggest that the information transfer from the third party, via the developer, to the user is susceptible to omissions and mistakes. We believe that our system can assist app developers, which we tested with a major European electronic device manufacturer in order to help them identify the privacy practices of various legacy apps (§ 7).

5.4 What Characteristics of Apps Are Associated with Potential Compliance Issues?

Our system downloads the Play Store metadata associated with each app. This metadata describes various characteristics of apps, such as the date the app was last updated, the Play Store categories the app belongs to, and the Entertainment Software Rating Board (ESRB) content rating of the app. In this section, we discuss how different app characteristics are associated with potential compliance issues.

[23] See, e.g., Cal. Bus. & Prof. Code §22575(b)(6).

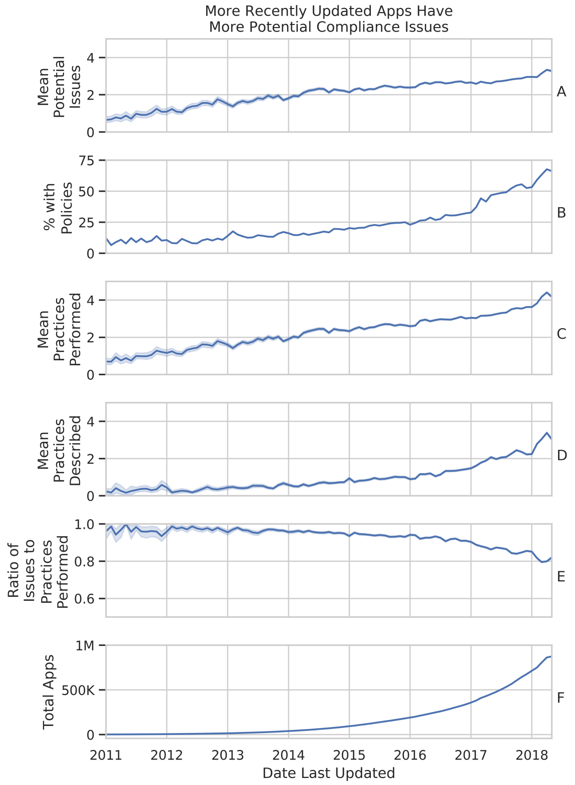


Fig. 9. We find that the mean number of potential issues is greater for more recently updated apps (A). The graphs cover January 2011, the start of the first year with at least 1,000 apps in our dataset, to May 2018, the month in which our analysis ran at scale. 95% confidence intervals are shown in lighter color.

More Recently Updated Apps Have More Potential Issues

For every app, our system retrieved the date when it was last updated (or, if it was never updated, when it was initially published). Figure 9 shows how more recently updated (newer) apps differ from less recently updated (older) apps. Despite newer apps being more likely to have privacy policies (Figure 9 B), they have more potential issues than older apps (Figure 9 A). Our analysis suggests that as newer apps perform more practices (Figure 9 C), they also provide more opportunities for potential issues to occur. The number of practices described in privacy policies has also increased (Figure 9 D); evidently, this increase has not kept pace with the simultaneous increase in practices performed. Thus, transparency is not guaranteed by apps simply having privacy policies. Rather, transparency requires that those policies comprehensively describe apps' practices.

Overall, our findings suggest that Google's efforts to compel developers to post privacy policies [8] may not suffice by itself. Without the proper tools and incentives, developers may not be able or willing to comprehensively describe their apps' behaviors. Conversely, we also observe positive developments. The ratio of the number of potential compliance issues to the number of practices performed is decreasing for newer apps (Figure 9 E), even as the total number of apps is increasing overall (Figure 9 F). App developers seem to be getting better at describing their apps' practices, though not enough to offset the fact that newer apps perform more practices.

Even Kids' Apps Have Potential Issues

Location-related information is particularly sensitive, yet our analysis found location-related potential issues to be relatively common (§ 5.3). Thus, we use our analysis results to gain greater insight into which apps are affected by these potential issues. Figure 10 shows a heatmap with the ratio of location-related potential issues to location-related practices performed, grouped by Play Store category. It can be observed that apps in certain categories have greater transparency than others. In particular, our results suggest a relatively low prevalence of potential issues in the FAMILY_ACTION and FAMILY_PRETEND categories. However, apps in FAMILY_CREATE and FAMILY_EDUCATION do not appear to be significantly better than the other store categories. Regardless, the detection of *any* potential issues in the FAMILY categories may subject apps to regulatory scrutiny (§ 6), as apps in those categories have to adhere to the heightened privacy requirements of COPPA. Google will only include apps in FAMILY categories if developers have affirmed vis-a-vis Google that their apps are COPPA compliant and eligible to participate in the Designed for Families program [18]. Potential issues of apps in these categories are in direct conflict with such affirmations.

Individual Developer Activity May Impact the Overall Number of Potential Issues

It is striking that apps in the BOOKS_AND_REFERENCE, COMICS, and LIBRARIES_AND_DEMO categories have particularly high ratios of performed practices to potential issues, for both first and third party practices. One reason may be that a large number of apps in these categories come from the same developers, who are using the same location APIs across their apps without dis-

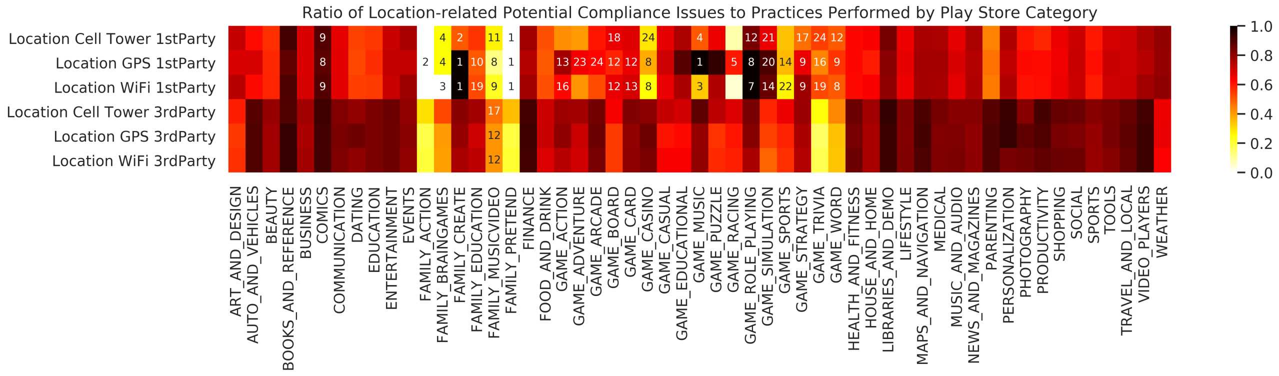


Fig. 10. The ratio of location-related potential compliance issues and practices performed. Lighter colors indicate greater transparency of practices. Darker colors indicate that practices are being performed but not disclosed. Cells with fewer than 25 apps performing the practice are annotated with the respective number of apps. Figure F3 in the Appendix has been extended to include all types of potential compliance issues.

closing it in their privacy policy (which is also the same for all their apps). For example, there are 1,104 apps in the BOOKS_AND_REFERENCE category that access cell tower location data as a first party. 58% (644/1,104) have at least one policy, which is, in fact, a higher percent than the mean for all apps in our dataset (§ 5.1). However, 23% (146/644) of those apps appear to come from the same developer, who supplies custom apps for trade group conferences and other venues, and whose apps seem to use location functionality without disclosing it in their privacy policy. This example illustrates the impact an individual developer can have on the overall state of privacy in segments of an app ecosystem. In these areas, targeted enforcement may be impactful (§ 6). In general, our results in Figure 10 once more highlight the need for providing transparency for third party practices, as they appear to be substantially more opaque compared to first party practices. Note that the set of third party libraries is diverse. It includes ad networks, analytics services, social networks, and developer tools. Arguably, some may be considered less privacy-invasive, such as developer tools.

“Unrated” Apps Have Poor Transparency

The ESRB content ratings [15] are indicators of the intended audience of an app; these content ratings also appear to be associated with the existence of potential compliance issues. Figure 11 displays the ratio of apps with potential issues to performed practices grouped by ESRB content rating. The apps shown as “Unrated” on the Play Store have dramatically lower rates of practice disclosure. A similar pattern holds for almost all other practices (Figure F2 in the Appendix). Apps are shown as “Unrated” on the Play Store if their developers have not completed a questionnaire describing their apps’ age appropriateness.

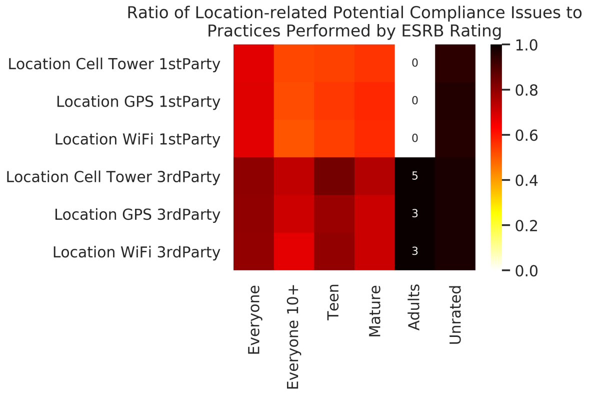
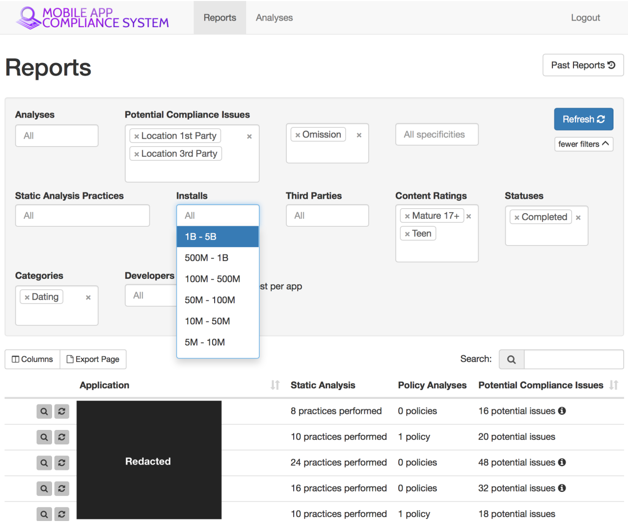


Fig. 11. The ratio of apps with location-related potential issues to practices performed grouped by ESRB content rating. ESRB ratings describe an app’s appropriateness for different age groups and are determined based on a questionnaire completed by app developers. Lighter colors indicate greater transparency of practices. Darker colors indicate that practices are being performed but not disclosed. Cells with fewer than 25 apps performing the practice are annotated with the respective number of apps. Figure F2 in the Appendix has been extended to include all types of potential compliance issues.

Despite Google announcing their plan to remove “Unrated” apps in 2015 [11], at the time of our analysis in April through May 2018 there were still 43,533 such apps on the Play Store. However, as of August 2, 2018, 91.8% of these apps were no longer available, 6.8% were still present but now had ESRB ratings, and 1.3% were still present but without ESRB ratings. These results suggest that Google intensified the removal of apps



Reports

Analyses: All

Potential Compliance Issues: × Location 1st Party, × Location 3rd Party; × Omission; All specificities

Static Analysis Practices: All

Installs: All

- 1B - 5B
- 500M - 1B
- 100M - 500M
- 50M - 100M
- 10M - 50M
- 5M - 10M

Third Parties: All

Content Ratings: × Mature 17+, × Teen

Statuses: × Completed

Categories: × Dating

Developers: All

st per app

Search:

Application	Static Analysis	Policy Analyses	Potential Compliance Issues
Redacted	8 practices performed	0 policies	16 potential issues ℹ
	10 practices performed	1 policy	20 potential issues
	24 practices performed	0 policies	48 potential issues ℹ
	16 practices performed	0 policies	32 potential issues ℹ
	10 practices performed	1 policy	18 potential issues

Fig. 12. Our system's user interface allows the user to filter apps according to various criteria, such as specific types of privacy practices which the app engages in, or the specific third party libraries the app accesses. The names of the apps have been redacted.

without ESRB ratings during this time period, which may have had the side effect of increasing the transparency of privacy practices on the Play Store overall. The results also indicate that Google is willing to remove apps which do not adhere to its developer policies, albeit with some delay.

6 Supporting Regulators

The number of mobile apps published in app stores, their complexity, and the many different third party libraries with which they can interface, make it impossible for regulators to comprehensively check all mobile apps for potential compliance issues. This problem is further compounded by the frequency at which apps are updated. A scalable system for identifying potential compliance issues offers the promise of changing this situation. Our web-based system provides a user interface (Figure 12) that enables users at regulatory agencies to filter analysis results according to a number of criteria, e.g., Play Store category. This functionality has been piloted with several regulatory agencies. Most recently, we interacted with personnel at the FTC to focus on potential compliance issues under COPPA (§ 5.4). We used our interface to impose selective criteria to identify code that could allow the collection of personal information, and our analysis zoomed in on a set of 9 popular apps from Google's Designed for Families program, a program under which app developers affirm to Google that their apps are COPPA compliant [18].

Based on the selected criteria, our automatic analysis identified 60 practices performed across the 9 apps. We manually verified the app analysis as before (§ 4.4) and found 47 true positives, 10 false positives, and 0 false negatives resulting in a precision of 82%, a recall of 100%, and an F1 score of 90%. We also observed 3 unknown instances. Legal experts reviewed the content of each privacy policy and confirmed our system's interpretation of relevant statements (or absence thereof) in the apps' privacy policies. Some of the potential compliance issues were easy to spot. For example, the privacy policy of one app was only one line, stating that it does not collect or disclose any personal information. In fact, our system's analysis as well as our manual verification showed that third party code within this app accesses cookies, device IDs, and mobile carrier identifiers. However, our system is designed to detect data access, which does not necessarily mean that the data leaves the device (§ 4.4). For a COPPA violation to occur we need evidence of data leaving the device. This study nevertheless shows the value of our static analysis, as it can be used to quickly zoom in on a small number of apps with potential compliance issues. Determining for apps identified in this manner whether data is actually transferred off the device could then be done using dynamic analysis tools [13, 45]. Because dynamic analysis of this type is computationally intensive, MAPS can significantly reduce processing requirements by quickly zooming in on apps with potential compliance issues and limiting the use of dynamic analysis to only those apps.

7 Supporting App Developers

While it is our objective to support regulatory agencies and app store operators, another important goal is to support app developers as they check their apps for potential compliance issues. Many apps are developed by one or two developers who often lack the expertise and resources necessary to properly disclose their app's privacy practices [3]. Larger outfits can also benefit from compliance tools as they check for compliance with privacy regulations. In particular, our analysis shows that developers often struggle to identify privacy practices associated with third party libraries (§ 5.3). With legislation such as the GDPR, which has opened the door to steeper penalties for privacy compliance violations, tools that can help organizations spot potential compliance issues can be expected to become increasingly important.

We piloted our system with a large European electronic device manufacturer to evaluate its usefulness in helping a sophisticated organization identify GDPR-related potential issues in some of its mobile apps. Apps selected for this study included both apps developed inhouse and an app that the company added to its lineup as a result of an acquisition. The company no longer had access to the developers who had originally developed one of the apps. Our system was used to initiate a due diligence process, helping focus manual efforts on problems it automatically identified as potential issues. The system also provided a basis for developing a comprehensive methodology for GDPR compliance analysis, which included additional potential compliance issues our system is not able to detect. Because our system is not perfect, all potential issues were manually checked. While a more comprehensive study would need to be conducted to accurately evaluate the usefulness of our system, anecdotal evidence suggests that it enabled the compliance team to speed up its review process and helped to quickly identify various potential issues that required careful attention.

8 Conclusions

In this study we introduced MAPS, a distributed system for assessing the state of privacy in the Android ecosystem at app store-wide scale. Our results from analyzing 1,035,853 free apps on the Play Store suggest that privacy can be improved for large numbers of apps, particularly when it comes to the disclosure of third party practices. Notice and choice is an elementary building block of Internet privacy. However, while privacy policies are intended to disclose applicable privacy practices, they are often incongruous with the actual practices performed.

In the app ecosystem, app stores are particularly well positioned to act as gatekeepers for the evaluation of whether the apps they are hosting are privacy-compliant. We believe that approaches like ours can help them to systematically and consistently analyze and improve privacy compliance. Further, it is the role of privacy regulators to identify problematic privacy practices and uphold privacy laws. However, with limited resources they have not been able to keep up with the increasing number of apps exhibiting potentially non-compliant practices. Our system has the potential to change this unsatisfactory situation: regulators can quickly zoom in on suspicious apps based on various filtering criteria and investigate potential compliance issues. Taken together, our different pilot studies suggest that our system can add value in different contexts — both at scale and in more focused tasks as piloted with the FTC for COPPA and with an electronic device manufacturer for the GDPR.

It is becoming increasingly clear that large platforms — whether they are app stores or other services — are not sustainable without sufficiently protecting their users' privacy. As privacy legislation is evolving, it is desirable to have technologies that can help with enforcement too. We understand our system as a contribution towards scaling privacy protection. In this study, we shed light on some of the most pressing privacy questions. However, substantial future work remains. Various stakeholders provided valuable feedback that we hope to incrementally incorporate into our system. For example, we hope to extend our system with functionality for detecting data processing via side-channels. We also would like to perform repeated analyses of app populations and run analyses on other Android app stores.

Acknowledgments

We thank our shepherd Mobin Javed as well as Jongkyu Talan Baek, Sushain Cherivirala, Roger Iyengar, Pingshan Li, Shaoyan Sam Li, Kanthashree Mysore Sathyendra, Florian Schaub, Xi Sun, and Shomir Wilson. This study was supported in part by the NSF Frontier grant on Usable Privacy Policies (CNS-1330596, CNS-1330141, and CNS-1330214) and a DARPA Brandeis grant on Personalized Privacy Assistants (FA8750-15-2-0277). The US Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF, DARPA, or the US Government. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing high performance computing resources that have contributed to the research results reported within this paper.

References

[1] V. Afonso, A. Bianchi, Y. Fratantonio, A. Doupe, M. Polino, P. de Geus, C. Kruegel, and G. Vigna, "Going native: Using a large-scale analysis of android apps to create a practical native-code sandboxing policy," in *NDSS '16*, Feb. 2016.

[2] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel, "FlowDroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps," *SIGPLAN Not.*, vol. 49, no. 6, pp. 259–269, Jun. 2014.

[3] R. Balebako, A. Marsh, J. Lin, J. Hong, and L. F. Cranor, "The privacy and security behaviors of smartphone app developers," in *USEC '14*, 2014.

[4] S. Bird, E. Klein, and E. Loper, "Natural language processing with python," 2014, accessed: June 19, 2019. [Online]. Available: http://www.nltk.org/book/ch11.html

[5] J. Bowers, B. Reaves, I. N. Sherman, P. Traynor, and K. R. B. Butler, "Regulators, mount up! Analysis of privacy policies for mobile money services," in *SOUPS '17*, 2017.

[6] California Department of Justice, "Attorney General Kamala D. Harris secures global agreement to strengthen privacy protections for users of mobile applications," http://www.oag.ca.gov/news/press-releases/attorney-general-kamala-d-harris-secures-global-agreement-strengthen-privacy, Feb. 2012, accessed: June 19, 2019.

[7] Y. Chen, W. You, Y. Lee, K. Chen, X. Wang, and W. Zou, "Mass discovery of android traffic imprints through instantiated partial execution," in *CCS '17*, 2017.

[8] B. Clark. (2017, Feb.) Millions of apps could soon be purged from Google Play Store. https://thenextweb.com/google/2017/02/08/millions-apps-soon-purged-google-play-store/.

[9] A. Continella, Y. Fratantonio, M. Lindorfer, A. Puccetti, A. Zand, C. Kruegel, and G. Vigna, "Obfuscation-resilient privacy leak detection for mobile apps through differential analysis," in *NDSS '17*, 2017.

[10] L. F. Cranor, P. G. Leon, and B. Ur, "A large-scale evaluation of U.S. financial institutions standardized privacy notices," *ACM Trans. Web*, vol. 10, no. 3, pp. 17:1–17:33, Aug. 2016.

[11] Don Reisinger, "Google Play gets serious with 'expert' screening, age ratings for Android apps," https://www.cnet.com/news/google-play-adds-app-ratings-to-inform-users-on-content/, Mar. 2015, accessed: June 19, 2019.

[12] B. Efron, "Bootstrap methods: Another look at the jackknife," in *Breakthroughs in statistics*. Springer, 1992, pp. 569–593.

[13] W. Enck, P. Gilbert, 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," in *OSDI '10*, 2010.

[14] T. Ermakova, B. Fabian, and E. Babina, "Readability of privacy policies of healthcare websites," in *Wirtschaftsinformatik '15*, 2015.

[15] ESRB, "ESRB ratings guide," http://www.esrb.org/ratings/ratings_guide.aspx, 2018, accessed: June 19, 2019.

[16] FTC, "Complaint Path," https://www.ftc.gov/sites/default/files/documents/cases/2013/02/130201pathinccmpt.pdf, Feb. 2013, accessed: June 19, 2019.

[17] C. Gibler, J. Crussell, J. Erickson, and H. Chen, "AndroidLeaks: Automatically detecting potential privacy leaks in android applications on a large scale," in *TRUST '12*, 2012.

[18] Google, "Designed for families addendum," https://play.google.com/intl/ALL_us/about/families/developer-distribution-agreement-addendum.html, 2015, accessed: June 19, 2019.

[19] Google, "Google analytics terms of service," https://www.google.com/analytics/terms/us.html, 2018, accessed: June 19, 2019.

[20] ——, "Google developer policy center user data," https://play.google.com/about/privacy-security-deception/user-data/, 2018, accessed: June 19, 2019.

[21] Google, "Play console help," https://support.google.com/googleplay/android-developer/answer/6048248?hl=en, 2018, accessed: June 19, 2019.

[22] M. I. Gordon, D. Kim, J. Perkins, L. Gilham, N. Nguyen, and M. Rinard, "Information-flow analysis of android applications in DroidSafe," in *NDSS '15*, 2015.

[23] H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, and K. Aberer, "Polisis: Automated analysis and presentation of privacy policies using deep learning," in *USENIX Security '18*, 2018.

[24] J. Huang, O. Schranz, S. Bugiel, and M. Backes, "The art of app compartmentalization: Compiler-based library privilege separation on stock android," in *CCS '17*, 2017.

[25] L. Lei, Y. He, K. Sun, J. Jing, Y. Wang, Q. Li, and J. Weng, "Vulnerable implicit service: A revisit," in *CCS '17*, 2017.

[26] T. Libert, "An automated approach to auditing disclosure of third-party data collection in website privacy policies," in *WWW '18*, 2018.

[27] J. Lin, B. Liu, N. Sadeh, and J. I. Hong, "Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings," in *SOUPS '14*. USENIX Assoc., 2014.

[28] B. Liu, B. Liu, H. Jin, and R. Govindan, "Efficient privilege de-escalation for ad libraries in mobile apps," in *MobiSys '15*, 2015.

[29] F. Liu, S. Wilson, P. Story, S. Zimmeck, and N. Sadeh, "Towards automatic classification of privacy policy text," School of Computer Science Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-ISR-17-118R and CMU-LTI-17-010, Jun. 2018.

[30] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.

[31] E. Mariconti, L. Onwuzurike, P. Andriotis, E. D. Cristofaro, G. J. Ross, and G. Stringhini, "Mamadroid: Detecting android malware by building markov chains of behavioral models," in *NDSS '17*, 2017.

[32] F. Marotta-Wurgler, "Does "notice and choice" disclosure regulation work? An empirical study of privacy policies," https://www.law.umich.edu/centersandprograms/lawandeconomics/workshops/Documents/Paper13.Marotta-Wurgler.Does%20Notice%20and%20Choice%20Disclosure%20Work.pdf, 2015, accessed: June 19, 2019.

[33] A. M. McDonald and L. F. Cranor, "The cost of reading privacy policies," *I/S: A Journal of Law and Policy for the Information Society*, vol. 4, no. 3, pp. 540–565, 2008.

[34] P. Mutchler, A. Doupé, J. Mitchell, C. Kruegel, and G. Vigna, "A large-scale study of mobile web app security," in *MoST '15*, 2015.

[35] Y. Nan, Z. Yang, X. Wang, Y. Zhang, D. Zhu, and M. Yang, "Finding clues for your secrets: Semantics-driven, learning-

based privacy discovery in mobile apps," in *NDSS '17*, 2017.

[36] R. Neisse, G. Steri, D. Geneiatakis, and I. N. Fovino, "A privacy enforcing framework for android applications," *Computers & Security*, vol. 62, pp. 257 – 277, 2016.

[37] Oracle, "Naming a package," https://docs.oracle.com/javase/tutorial/java/package/namingpkgs.html, 2017, accessed: June 19, 2019.

[38] X. Pan, X. Wang, Y. Duan, X. Wang, and H. Yin, "Dark hazard: Learning-based, large-scale discovery of hidden sensitive operations in android apps," in *NDSS '17*, 2017.

[39] R. Ramanath, F. Liu, N. Sadeh, and N. A. Smith, "Unsupervised alignment of privacy policies using hidden markov models," in *ACL '14*, 2014.

[40] 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 '18*, 2018.

[41] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman, and V. Paxson, "Haystack: In situ mobile traffic analysis in user space," *CoRR*, vol. abs/1510.01419, 2015.

[42] D. Reidsma and J. Carletta, "Reliability measurement without limits," *Comput. Linguist.*, vol. 34, no. 3, pp. 319–326, Sep. 2008.

[43] 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 '18*, 2018.

[44] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes, "Recon: Revealing and controlling PII leaks in mobile network traffic," in *MobiSys '16*, 2016.

[45] I. Reyes, P. Wijesekera, J. Reardon, A. E. B. On, A. Razaghpanah, N. Vallina-Rodriguez, and S. Egelman, ""Won't somebody think of the children?" Examining COPPA compliance at scale," in *PETS '18*, vol. 3, 2018, pp. 63–83.

[46] N. Sadeh, A. Acquisti, T. D. Breaux, L. F. Cranor, A. M. McDonald, J. R. Reidenberg, N. A. Smith, F. Liu, N. C. Russell, F. Schaub, and S. Wilson, "The usable privacy policy project," Carnegie Mellon University, Tech. report CMU-ISR-13-119, 2013.

[47] K. M. Sathyendra, S. Wilson, F. Schaub, S. Zimmeck, and N. Sadeh, "Identifying the provision of choices in privacy policy text," in *EMNLP '17*, 2017.

[48] scikit-learn developers, "sklearn.feature_extraction.text.tfidfvectorizer," http://scikit-learn.org/0.18/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html, 2016, accessed: June 19, 2019.

[49] ——, "sklearn.linear_model.logisticregression," http://scikit-learn.org/0.18/modules/generated/sklearn.linear_model.LogisticRegression.html, 2016, accessed: June 19, 2019.

[50] ——, "sklearn.svm.svc," http://scikit-learn.org/0.18/modules/generated/sklearn.svm.SVC.html, 2016, accessed: June 19, 2019.

[51] R. Slavin, X. Wang, M. Hosseini, W. Hester, R. Krishnan, J. Bhatia, T. Breaux, and J. Niu, "Toward a framework for detecting privacy policy violation in android application code," in *ICSE '16*, 2016.

[52] D. J. Solove and W. Hartzog, "The FTC and the new common law of privacy," *Columbia Law Review*, vol. 114, pp. 583–676, 2014.

[53] P. Story, S. Zimmeck, A. Ravichander, D. Smullen, Z. Wang, J. Reidenberg, N. C. Russell, and N. Sadeh, "Natural language processing for mobile app privacy compliance," *AAAI Spring Symposium on Privacy-Enhancing Artificial Intelligence and Language Technologies*, Mar. 2019.

[54] P. Story, S. Zimmeck, and N. Sadeh, "Which apps have privacy policies?" in *APF '18*, 2018.

[55] W. B. Tesfay, P. Hofmann, T. Nakamura, S. Kiyomoto, and J. Serna, "I read but don't agree: Privacy policy benchmarking using machine learning and the EU GDPR," in *WWW '18*, 2018.

[56] G. Tottie, *Negation in English speech and writing*. Academic Press, 1991.

[57] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, "XSEDE: Accelerating scientific discovery," *Computing in Science & Engineering*, vol. 16, no. 5, pp. 62–74, Sep. 2014.

[58] G. S. Tuncay, S. Demetriou, K. Ganju, and C. A. Gunter, "Resolving the predicament of android custom permissions," in *NDSS '18*, 2018.

[59] N. Viennot, E. Garcia, and J. Nieh, "A measurement study of Google Play," in *SIGMETRICS '14*, 2014.

[60] H. Wang, Z. Liu, Y. Guo, X. Chen, M. Zhang, G. Xu, and J. Hong, "An explorative study of the mobile app ecosystem from app developers' perspective," in *WWW '17*, 2017.

[61] X. Wang, X. Qin, M. B. Hosseini, R. Slavin, T. D. Breaux, and J. Niu, "GUILeak: Identifying privacy practices on GUI-based data," https://pdfs.semanticscholar.org/ced1/313acaacd3897b5b231cdccb1383d01d20c4.pdf, 2017, accessed: June 19, 2019.

[62] T. Watanabe, M. Akiyama, T. Sakai, and T. Mori, "Understanding the inconsistencies between text descriptions and the use of privacy-sensitive resources of mobile apps," in *SOUPS '15*, 2015.

[63] S. Wilson, F. Schaub, A. A. Dara, F. Liu, S. Cherivirala, P. G. Leon, M. S. Andersen, S. Zimmeck, K. M. Sathyendra, N. C. Russell, T. B. Norton, E. Hovy, J. Reidenberg, and N. Sadeh, "The creation and analysis of a website privacy policy corpus," in *ACL '16*, 2016.

[64] L. Yu, X. Luo, X. Liu, and T. Zhang, "Can we trust the privacy policies of android apps?" in *DSN '16*, 2016.

[65] Y. Zhuang, A. Rafetseder, Y. Hu, Y. Tian, and J. Cappos, "Sensibility Testbed: Automated IRB policy enforcement in mobile research apps," in *HotMobile '18*, 2018.

[66] S. Zimmeck and S. M. Bellovin, "Privee: An architecture for automatically analyzing web privacy policies," in *USENIX Security '14*, 2014.

[67] S. Zimmeck, Z. Wang, L. Zou, R. Iyengar, B. Liu, F. Schaub, S. Wilson, N. Sadeh, S. M. Bellovin, and J. Reidenberg, "Automated analysis of privacy requirements for mobile apps," in *NDSS '17*, 2017.

9 Appendix

9.1 Inter-annotator Agreement

Table T1 shows the inter-annotator agreement per practice among three annotators for 35 randomly selected policies in the APP-350 corpus. We calculated annotator agreement at the level of each privacy policy. Our measured agreement level is generally higher in comparison to previously reported results [67]. To the extent that the annotated practices are comparable, we hypothesize that our uniform approach of instructing and training the annotators could be a factor in the higher agreement levels.

Policy Annotation	K's α	+/-/+-/0 Support
Contact 1stParty	0.75	26/0/0/79
Contact 3rdParty	0.65	5/0/1/99
Contact Email Address 1stParty	0.70	82/7/5/11
Contact Email Address 3rdParty	0.80	12/3/2/88
Contact Phone Number 1stParty	0.96	67/6/0/32
Contact Phone Number 3rdParty	0.92	9/3/2/91
Identifier 1stParty	0.63	38/0/0/67
Identifier 3rdParty	0.55	19/0/0/86
Identifier Cookie 1stParty	0.81	81/3/0/24
Identifier Cookie 3rdParty	0.76	67/0/1/37
Identifier Device ID 1stParty	0.69	59/0/0/46
Identifier Device ID 3rdParty	0.76	29/0/0/76
Identifier IMEI 1stParty	0.94	20/0/0/85
Identifier IMEI 3rdParty	0.79	5/0/0/100
Identifier IMSI 1stParty	0.90	11/0/0/94
Identifier IMSI 3rdParty	1.00	3/0/0/102
Identifier MAC 1stParty	0.73	24/0/0/81
Identifier MAC 3rdParty	0.89	10/0/0/95
Identifier Mobile Carrier 1stParty	0.82	21/0/0/84
Identifier Mobile Carrier 3rdParty	0.76	9/0/0/96
Identifier SIM Serial 1stParty	0.65	17/0/0/88
Identifier SIM Serial 3rdParty	0.49	2/0/0/103
Identifier SSID BSSID 1stParty	1.00	3/0/0/102
Identifier SSID BSSID 3rdParty	1.00	0/0/0/105
Location 1stParty	0.67	58/9/3/35
Location 3rdParty	0.71	28/5/0/72
Location Cell Tower 1stParty	0.66	22/0/0/83
Location Cell Tower 3rdParty	0.89	7/3/0/95
Location GPS 1stParty	0.85	29/8/0/68
Location GPS 3rdParty	0.92	9/4/0/92
Location WiFi 1stParty	0.77	22/6/0/77
Location WiFi 3rdParty	0.80	7/4/0/94
Single Sign On	0.64	41/0/0/64
Single Sign On: Facebook	0.78	33/0/0/72

Table T1. Inter-annotator agreement as measured with Krippendorff's α (K's α). The Support column (+/-/+-/0 Support) shows the sum of all three annotators' positive (+), negative (-), positive and negative (+-), and silent (0) annotations, respectively.

9.2 Android APIs

Table T2 shows the Android APIs used in our system's static analysis module (§ 4.4) to identify the privacy practices an app performs. We included some of the most frequently used APIs as indicated by their prevalence on publicly accessible GitHub repositories.

Privacy Practice	APIs
Contact Email Address	android.accounts.AccountManager.getAccounts android.accounts.AccountManager.getAccountsByType android.accounts.AccountManager.getAccountsByTypeAndFeatures android.accounts.AccountManager.getAccountsByTypeForPackage android.accounts.AccountManager.hasFeatures android.net.MailTo.getTo android.net.MailTo.getCc
Contact Phone Number	android.telephony.TelephonyManager.getLine1Number
Identifier Cookie	android.webkit.CookieManager.getInstance
Identifier Device ID	android.provider.Settings.Secure.getString
Identifier IMEI	android.telephony.TelephonyManager.getDeviceId android.telephony.TelephonyManager.getImei
Identifier IMSI	android.telephony.TelephonyManager.getSubscriberId
Identifier MAC	android.net.wifi.WifiInfo.getMacAddress
Identifier Mobile Carrier	android.telephony.TelephonyManager.getNetworkOperator android.telephony.TelephonyManager.getNetworkOperatorName android.telephony.TelephonyManager.getSimOperator android.telephony.TelephonyManager.getSimOperatorName
Identifier SIM Serial	android.telephony.TelephonyManager.getSimSerialNumber
Identifier SSID BSSID	android.net.wifi.WifiManager.getConfiguredNetworks android.net.wifi.WifiInfo.getBSSID android.net.wifi.WifiInfo.getSSID
Location Cell Tower	FusedLocationProviderClient.getLastLocation android.location.LocationManager.requestLocationUpdates android.location.LocationManager.requestSingleUpdate android.location.LocationManager.getLastKnownLocation android.location.LocationManager.addProximityAlert android.telephony.gsm.GsmCellLocation.getCid android.telephony.gsm.GsmCellLocation.getLac android.telephony.TelephonyManager.getCellLocation android.telephony.TelephonyManager.getAllCellInfo android.telephony.TelephonyManager.getNeighboringCellInfo
Location GPS and WiFi	FusedLocationProviderClient.getLastLocation android.location.LocationManager.requestLocationUpdates android.location.LocationManager.requestSingleUpdate android.location.LocationManager.getLastKnownLocation android.location.LocationManager.addProximityAlert

Table T2. Android APIs used in our system's static analysis module. For each privacy practice it is sufficient that one API is detected.

9.3 Confidence Intervals for Identifying Potential Compliance Issues

We conducted a statistical analysis of our techniques for detecting potential compliance issues in order to determine the effect of their error rates (§ 4.5, Table 3) on our large-scale analysis results (§ 5). Our goal was to quantify the extent to which we may over- or undercount potential compliance issues due to the error rates of our techniques. We used bootstrap statistics [12] to estimate the true percent of potential compliance issues for each practice. Specifically, for each practice, we randomly resample with replacement from our test set. For each resampling, we calculate an estimate of the percent

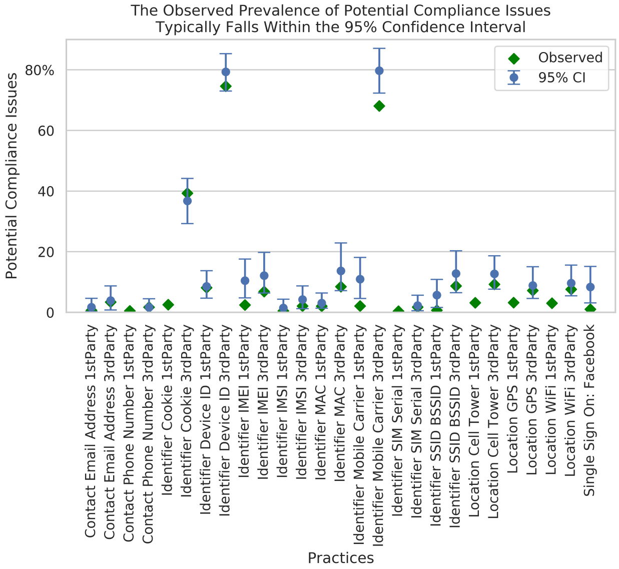


Fig. F1. The observed prevalence of potential compliance issues generally falls in or below the 95% confidence interval for each practice. Thus, we do not find evidence that we overestimate the number of potential compliance issues, on average.

of potential compliance issues with:

$$\big(P(\text{TP}|\text{TP or FP}) * \text{Positive Observed}$$
$$+ P(\text{FN}|\text{FN or TN}) * \text{Negative Observed}\big)$$
$$\div \text{ Total Observed} * 100$$

where TP, FP, TN, and FN denote true positives, false positives, true negatives, and false negatives, respectively, in the resample of the test set. “Observed” refers to the number of observed positive, negative, and total instances in our large-scale analysis.

After 1,000 resamplings and recalculations of the estimate, the 2.5 and 97.5 percentiles of the bootstrapped estimates formed the 95% confidence interval of the estimates. One caveat to this procedure is that for practices with few positive instances, the estimate can be undefined, due to division by zero when calculating the conditional probabilities. In cases where fewer than 5% of the 1,000 bootstrapped estimates were undefined, we simply discarded the undefined estimates. If 5% or more estimates were undefined, we did not calculate a confidence interval for the practice.

Comparing the confidence intervals to our observed values, as depicted in Figure F1, the observed prevalence of potential compliance issues generally falls in or below the 95% confidence interval. This result suggests that our results underestimate the number of potential compliance issues for some practices. However, we find no evidence that our results overestimate the number of potential compliance issues, on average. The confidence intervals support two of our main conclusions. First, various identifier- and location-related potential compliance issues are prevalent. Second, third party potential compliance issues are generally more prevalent than first party potential compliance issues.

9.4 Ratio of Potential Compliance Issues to Practices Performed by ESRB Content Rating

Figure F2 shows the ratio of apps with potential issues to performed practices grouped by ESRB content rating.

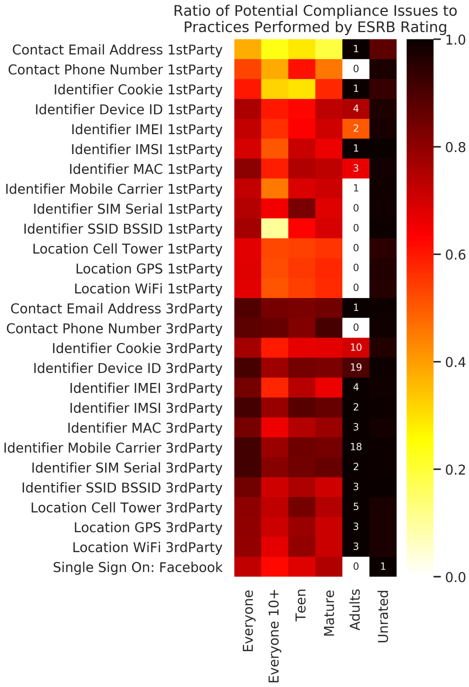


Fig. F2. Lighter colors indicate greater transparency of practices. Darker colors indicate that practices are being performed but not disclosed. Cells with fewer than 25 apps performing the practice are annotated with the respective number of apps.

9.5 Ratio of Potential Compliance Issues to Practices Performed by Play Store Category

Figure F3 shows the ratio of apps with potential issues to performed practices grouped by Play Store category.

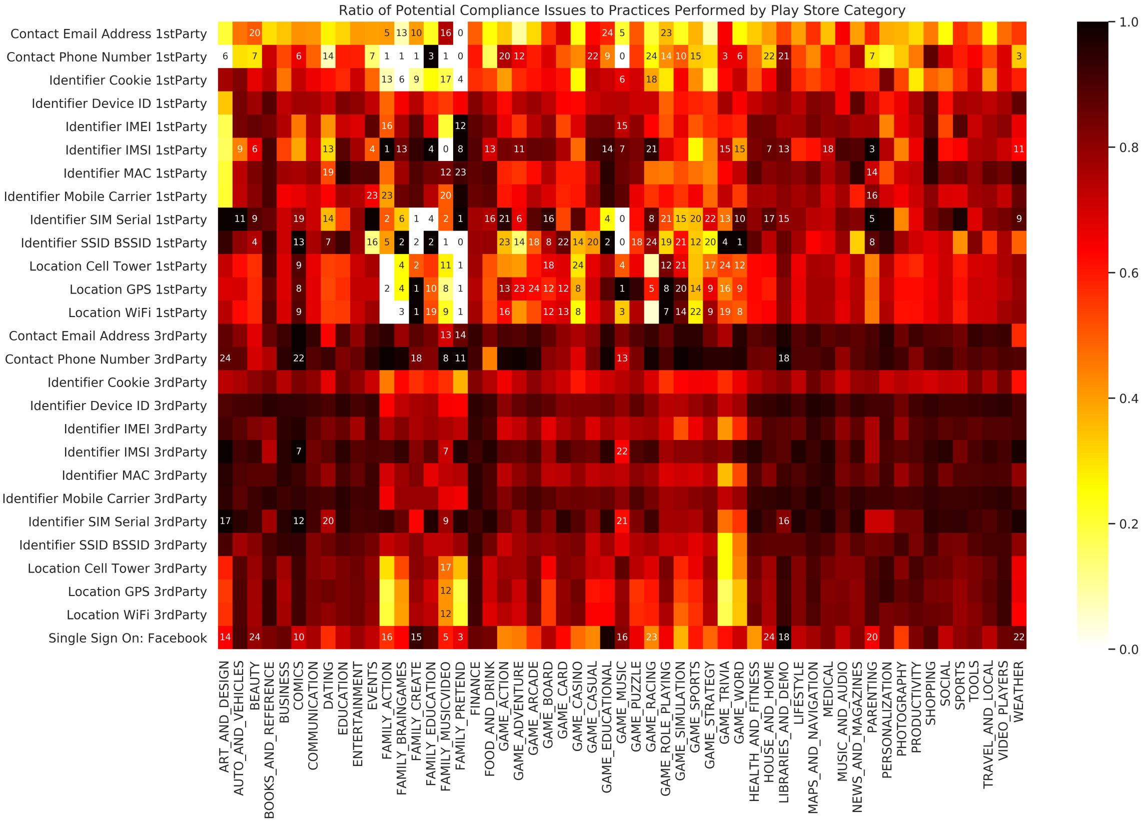


Fig. F3. Lighter colors indicate greater transparency of practices. Darker colors indicate that practices are being performed but not disclosed. Cells with fewer than 25 apps performing the practice are annotated with the respective number of apps.