Understanding the Evolution of Mobile App Ecosystems: A Longitudinal Measurement Study of Google Play

Haoyu Wang
Beijing University of Posts and Telecommunications
Beijing, China
haoyuwang@bupt.edu.cn

Hao Li
OrangeAPK, Inc.
Beijing, China
lihao_0823@yeah.net

Yao Guo
MOE Key Lab of HCST, School of EECS, Peking University
Beijing, China
yaoguo@pku.edu.cn

ABSTRACT

The continuing expansion of mobile app ecosystems has attracted lots of efforts from the research community. However, although a large number of research studies have focused on analyzing the corpus of mobile apps and app markets, little is known at a comprehensive level on the evolution of mobile app ecosystems. Because the mobile app ecosystem is continuously evolving over time, understanding the dynamics of app ecosystems could provide unique insights that cannot be achieved through studying a single static snapshot. In this paper, we seek to shed light on the dynamics of mobile app ecosystems. Based on 5.3 million app records (with both app metadata and apks) collected from three snapshots of Google Play over more than three years, we conduct the first study on the evolution of app ecosystems from different aspects. Our results suggest that although the overall ecosystem shows promising progress in regard of app popularity, user ratings, permission usage and privacy policy declaration, there still exists a considerable number of unsolved issues including malicious apps, update issues, third-party tracking threats, improper app promotion behaviors, and spamming/malicious developers. Our study shows that understanding the evolution of mobile app ecosystems can help developers make better decision on developing and releasing apps, provide insights for app markets to identifying misbehaviors, and help mobile users to choose desired apps.

CCS CONCEPTS

• **Information systems** → **Web mining**; • **Security and privacy** → **Mobile and wireless security**; • **Networks** → **Mobile networks**; • **Software and its engineering** → *Software libraries and repositories.*

KEYWORDS

App ecosystem; Android; Google Play; evolution

ACM Reference Format:
Haoyu Wang, Hao Li, and Yao Guo. 2019. Understanding the Evolution of Mobile App Ecosystems: A Longitudinal Measurement Study of Google Play. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3308558.3313611

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313611

1 INTRODUCTION

It has been over 10 years since the first Android-based smartphone was announced in 2008 [18, 27]. Android has evolved significantly in the past decade, and has become the most popular mobile operating system with 85.9% market share as of 2017 [55]. Google reported that there were over 2 billion monthly active devices using Android as of May 2017 [48]. Millions of mobile apps with myriad features have become the key to boost the evolution of smartphones and Android. Google Play, the official Android app market, was first introduced in 2008 [72]. Since its launch, (Android) mobile apps have impacted our society at an astounding pace and scale. It is now one of the largest app stores in the world, with over 2.5 million apps available for download in October 2018[1]. The total number of app downloads in Google Play has surpassed 200 billion by 2017 [70].

Billions of mobile users, millions of mobile apps and app developers, thousands of devices manufacturers, and hundreds of app markets have formed a symbiotic relationship, which we refer to as the ***mobile app ecosystem***. The global app economy is also poised to explode, which is estimated to worth 6.3 trillion dollars by 2021 [45]. As a result, the continuing evolution of mobile app ecosystems has increased the complexity and myth of the whole ecosystem, which itself needs considerable efforts to understand its characteristics and working mechanism.

The mobile app ecosystem has attracted a lot of efforts from the research community. A large number of studies have focused on analyzing the mobile apps and app markets, including exploring the characteristics of mobile apps [11, 32, 35, 43, 59], understanding and comparing app markets [7, 14, 60, 67, 69, 70], security and privacy analysis of mobile apps [10, 25, 26, 41], understanding app developers and mobile users [16, 22, 33, 40, 74], etc.

However, little is known at a comprehensive level on the **evolution of mobile app ecosystems**. Although several previous work have performed large-scale measurement studies on the app ecosystem and analyzed millions of apps [60, 70], they only studied the ecosystem based on a static snapshot collected in a given time slot. For example, PlayDrone [60] performed the characterization of 1.1 million Google Play apps crawled in 2013, and Wang *et al.* [69] studied the ecosystem from the perspective of app developers on over 1.2 million Google Play apps and 320,000 developers collected in 2015. To the best of our knowledge, no previous work have studied the evolution of app ecosystems comprehensively at a large scale.

Because the Android ecosystem is evolving over time, studying the dynamics of the app ecosystem could gain unique insights

[1]It is reported that the number of apps in Google Play has surpassed 3 million in 2017. However, Google has been removing a significant number of apps from time to time.

that cannot be achieved by studying static app market snapshots. From app developers' perspective, understanding the evolution of app ecosystems could help them make better decision on developing and releasing apps, e.g., realizing which app categories are most competitive and the trend, and understanding the impact that developer actions will likely have on the success of their apps. From the view of app markets, studying the evolution of the app ecosystem could help them with monitoring and further improving the app market, as the number of apps keeps increasing. Previous studies suggest that malicious apps and fraudulent developers are prevalent in app markets [60, 70], thus lessons learned from a longitudinal analysis can provide insights into identifying fraudulent app promotion and malicious behaviors. Besides, a longitudinal measurement study of the ecosystem is a valuable indicator of understanding the real threats and challenges. From mobile users' perspective, understanding the dynamics of app markets could help them make better decision when choosing desired apps with similar functionality, e.g., avoiding the apps that have not been updated for years, which may contain vulnerabilities and have compatibility issues.

This Study. In this paper, we make the first effort towards understanding the evolution of mobile app ecosystems. Specifically, we seek to shed light on the evolution of Google Play, the official and the most popular Android app market. To this end, we have collected three snapshots of Google Play, which were crawled in March 2014, March 2015 and September 2017, respectively (Section 3). This dataset includes 5.3 million Android app entities in total, with all the metadata and apks. We first use this dataset to provide a high-level characterization of all the apps and analyze the evolving trends, including the distribution of free and paid apps, the evolution of app downloads and app rating, the ranking and competitiveness of app categories, the evolution of permission usage and privacy policy declaration, and the presence of third-party tracking and advertising libraries (Section 4). Next, for the 743,530 long-lasting apps (appeared in all three snapshots) in our dataset, we further investigate their growth rate, app updates, and permission changes (Section 5). Third, we provide an in-depth analysis of malicious and deceptive behaviors across the three snapshots, discussing the presence of malicious apps and aggressive app promotion behaviors (Section 6). At last, we analyze the evolution from the developers' perspective. We have classified the developers into different groups based on the number of apps they released and analyzed their evolution, and further pinpoint the spamming and malicious developers (Section 7). Among many interesting results and observations, the following are the most prominent:

- The app monetization scheme evolves during the evolution of the mobile app ecosystem. The number of paid apps has decreased significantly in Google Play, while mobile advertising is getting more popular.
- Although the popularity and overall quality of apps published in Google Play are getting better, the overall ecosystem shows a typical Pareto effect all the time, i.e., dominated by a small number of apps and developers.
- There exists a trend that the percentage of apps that request sensitive permissions are on the decline. Google's regulation on privacy policy has greatly improved the privacy protection of mobile users, e.g., the percentage of apps declaring privacy policies in 2017 is four times of the number in 2014.
- Over 61% of the long-lasting apps have not received any updates for three years, which indicates that these apps cannot utilize new features introduced by new Android versions, and may have compatibility issues and vulnerabilities.
- The proportion of potentially malicious apps in Google Play is decreasing. However, spamming apps that use misleading references to other apps are always prevalent in Google Play, many using more sophistic methods to insert keywords.
- Although Google Play removes spamming developers and malicious developers regularly, the number of developers with misbehaviors is nonetheless increasing.

To the best of our knowledge, this is the first comprehensive longitudinal study of the mobile app ecosystem at scale. Our dataset is by far one of the largest Google Play app repositories in the research community, with all the metadata and apks. We believe that our efforts and the revealed insights can benefit the research community, as well as various stakeholders of the mobile app ecosystem.

2 RELATED WORK

2.1 Mobile App Ecosystem Analysis

A number of research efforts have focused on analyzing the mobile app ecosystem [7, 28, 29, 31, 34, 38, 46, 60, 69, 70]. They usually first collect a large number of apps from Google Play or alternative markets, and then measure the app ecosystem from various dimensions. We have summarized the most related work on app ecosystem analysis, as shown in Table 1. Two representative studies are AndroidZoo [38] and the work of Wang *et al.* [70]. AndroZoo focused on crawling a large-scale dataset of APKs, which has enabled a number of studies focusing on malware analysis and app repository mining [31]. Wang *et al.* [70] performed a large-scale comparative study that covers more than 6 million Android apps downloaded from 16 alternative markets and Google Play to understand the catalog similarity across app stores and malicious behaviors. Although these studies have measured the mobile app ecosystem in large scale, they only captured a single snapshot of the ecosystem and did not consider the longitudinal evolution.

2.2 App Evolution Analysis

One line of research is analyzing the app evolution. Potharaju *et al.* [49] performed a longitudinal study of 160K apps, however, they only studied the evolution of metadata (e.g., price and downloads) and only covered 10% of apps in Google Play. Calciati *et al.* [13] and Taylor *et al.* [56, 57] analyzed the evolution of apps' permission requests, and found that apps tend to request more permissions in their evolution. Taylor and Martinovic [56] studied the evolution of financial apps from permission usage and vulnerabilities. Gao *et al.* [23] analyzed the vulnerability evolution of Android apps based on a set of Android app lineages in large scale, and found that apps do not get safer as they get updated. Wang *et al.* [67] found that a large portion of apps in Google Play are removed from time to time. They presented a large-scale study of 791,138 removed Google Play apps to identify potential reasons for app removal. Although these studies have analyzed app evolution from specific

Table 1: Related work on mobile app ecosystem analysis.

Paper	Market	# Apps (year)	Dimensions
PlayDrone [60]	Google Play	1,107,476 (2013)	Similar app, Library usage, Authentication schemes
Petsas et al. [46]	Five alternative markets	316,143 (2013)	Download patterns, popularity trends
Wang et al. [69]	Google Play	1,501,555 (2015)	Characterizing App Developers
Ali et al. [7]	Apple App Store, Google Play	80,000 pairs (2015)	App-store attributes Comparison
Ishii et al. [31]	Google Play, Alternative Markets	4,761,283 (2016)	Malware, Security Management
Want et al. [70]	Google Play, Chinese Markets	6,277,247 (2017)	Catalog Similarity, App Clones, Malware
AndroidZoo++ [38]	Google Play, Alternative Markets	Over 7 Million (2017)	App Repository

perspectives, these studies lack of a comprehensive understanding on the evolution of the app ecosystem.

2.3 Mobile App Analysis

A large number of studies focus on mobile app analysis, including security and privacy analysis [10, 25, 30, 39, 41, 42, 65, 66, 68], app repackaging detection [15, 63], app quality analysis [37, 50, 54, 58], third-party library detection [12, 44, 53, 62, 64] and mobile ad network analysis [17, 20], etc. Most of these studies focus on one specific issue and lack of the measurement study of the app ecosystem, although similar approaches could be used to understand the evolution of specific issues in the mobile app ecosystem.

3 STUDY DESIGN

3.1 Data Collection

We have created three snapshots of Google Play, which were crawled in 2014, 2015 and 2017, respectively. Note that we use the term **snapshot** to refer to the entire state of the market, i.e., it contains meta-information of (almost) all the apps and the corresponding apks. This dataset is representative enough to study the evolution of app ecosystem, as the timeline in our study period covers three major Android system updates (version 5.0, 6.0 and 7.0).

First Snapshot. It was created in March 2014. Our crawling strategy started from top 500 Google Play apps in each category (considered as seeds), and 12,500 apps that belong to 25 general categories in total. Then, we use a breadth-first-search approach to crawl (1) "Similar Apps" shown on the app web pages recommended by Google Play and (2) other apps released by the same developer by retrieving the developers' Google Play web pages. Note that we downloaded all the apk files of free apps through the Google Play API. Furthermore, We have taken efforts to crawl all the app metadata (e.g., app names, descriptions, version names, developer names, user ratings, app installs, the privacy policy links, etc). We have crawled over 1.5 million apps, which represent almost all the apps that can be crawled from Google Play at that time [60].

Second Snapshot. In March 2015 (1 year after the first snapshot), we repeated the same process as described above to crawl Google Play apps, except that we take the previous 1.5 million crawled apps as our searching seeds. It could ensure that we check all the 1.5 million apps crawled in 2014. Overall, we are able to collect more than 1.6 million Android apps in this snapshot.

Third Snapshot. We repeated the same process in September 2017 (2.5 years after the second snapshot), and we take the 1.6 million apps crawled in the second snapshot as searching seeds. At last, we have crawled 2.1 million apps in total.

Table 2: Summary of our dataset.

	# Apps	# Free	#Paid	Installs	# Developers
Google Play 2014	1,522,730	1,292,064	230,666	55.1B	413,933
Google Play 2015	1,650,895	1,408,593	242,302	90.2B	398,855
Google Play 2017	2,144,733	2,012,904	131,829	193.5B	541,105

We launched crawlers in parallel via Aliyun Cloud servers that were located in North America. In order to avoid potential regional/language bias, we instrumented our crawlers in the default mode that supports English. This is the largest longitudinal Android app dataset in our research community that has collected the detailed information of both apks and app metadata.

3.2 Dataset Characteristics

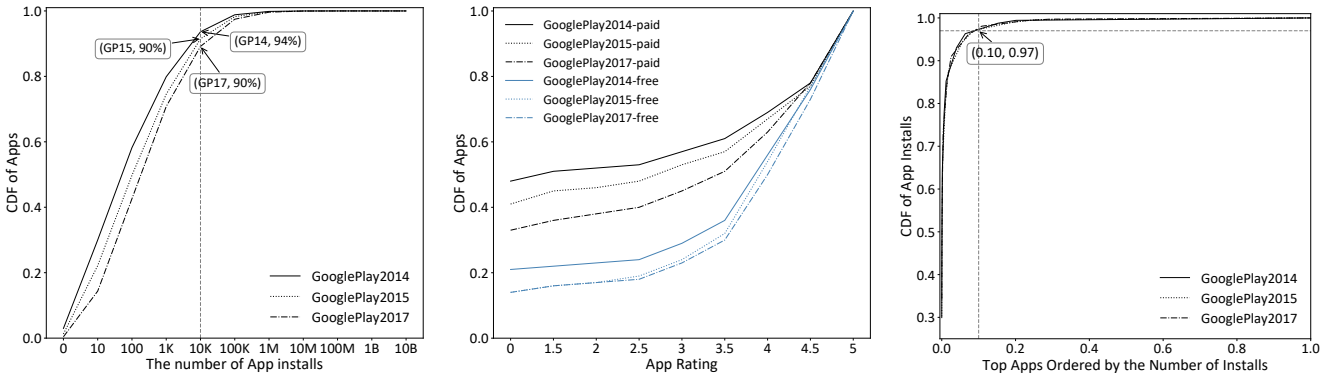
Table 2 reports the number of apps we harvested in the three snapshots. We could observe some general trends here: (1) From 2014 to 2017, the number of free apps in Google Play increases greatly, from roughly 1.3 million to the 2 million mark. In contrast, the number of paid apps decreases rapidly since 2015. The number of paid apps in 2017 is roughly half of the number in 2015. (2) The overall number of app installs in Google Play grows steadily. The number of app installs in 2015 is twice the number of 2014, while in 2017 the number has doubled compared to the 2015 snapshot. (3) The number of app developers in Google Play has not seen significant increase, which remains stable in the range of 400K to 500K. It is interesting to see that the number of app developers decrease in the 2015 snapshot compared to 2014.

4 OVERALL TREND OF APP EVOLUTION

4.1 Free vs. Paid Apps

It is a trend that the number of paid apps has decreased significantly in Google Play. The percentage of paid apps dropped by more than 60% (from 15.2% in 2014 to 6.1% in 2017). As to app installs, the proportion of paid apps is nearly negligible. Free apps occupy more than 99.7% of total app installs across snapshots.

There is a sharp decrease in the number of available paid apps from 2015 to 2017. To further investigate the reason, we analyzed the catalog similarity of paid apps between the three snapshots, and get two main observations: (1) More than 95% of the paid apps in 2017 are originated from 2015, while only 6,252 new paid apps have emerged in the two-year period. More than 99.5% of new apps crawled in the 2017 (1.28 million) snapshot are free apps. (2) Only 52% of paid apps (125,577) in 2015 remained as paid apps in 2017. For the remaining 116,725 apps, 6,989 of them (6%) became free apps (with the same app name and package name) in 2017, and roughly 94% of them had been removed/discontinued/renamed, so that we cannot locate them by searching the names in Google Play.



(a) The distribution of app installs. (b) The distribution of app ratings. (c) A few apps account for most of the downloads.

Figure 1: The evolution of app installs and app ratings.

Implication. One possible reason is that mobile ads and in-app purchase become the most popular monetization schemes in Google play apps (cf. Section 4.7). As a result, new developers should rethink their app releasing and monetization strategies.

4.2 App Installs

We further investigate the distribution of app installs across the three snapshots. Figure 1(a) reports the cumulative distribution of app installs, which suggests that ***the popularity and quality of apps published in Google Play is getting better***. Over 94% of apps have less than 10K installs by 2014, while the number is reduced to 89% by 2017. The average number of app installs was raised from 36,183 (2014) to 90,209 (2017).

Then we characterize the app popularity distribution for all the apps ordered by the number of installs, as illustrated in Figure 1(c). Our results highlight a typical **Pareto effect**, with top 0.1% of apps in Google Play represent roughly 60% of total installs, while the top 1% of apps account for 80% of total installs. This observation is consistent across all the three snapshots. It suggests that although the overall market shows a promising trend in regard of app popularity, the market is still dominated by a small percentage of apps.

4.3 App Ratings

We then investigate the distribution of app ratings in Google Play across the snapshots, as shown in Figure 1(b). In general, the distribution of app ratings for free apps is better than paid apps. The app ratings are getting better for both free and paid apps, i.e., more apps are getting higher user ratings. This trend is more obvious for paid apps. For the 2017 dataset, over 55% of paid apps received app ratings higher than 3-star, while the number in the 2014 dataset is 43%. For the free apps, the distribution of app ratings in the 2015 and 2017 dataset are almost the same, which are slightly better than in 2014.

4.4 App Categories

For the 2014 and 2015 snapshots, there are generally 25 categories, as listed in Figure 2. While, in 2017, 8 new categories were introduced, including "Beauty" and "Dating". As the distribution of apps and developers is fairly diverse for different categories, we examined the differences between each category in order to understand

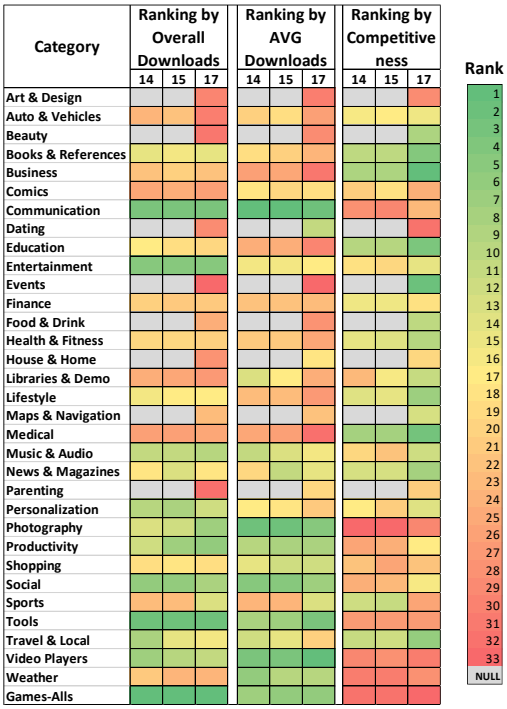


Figure 2: Distribution of app categories.

which categories are more popular and more competitive over time. This is also important to understand the trend of popular apps.

To this end, we compare the app categories based on (1) the total number of app installs, (2) the average number of app installs, and (3) the percentage of popular apps in each category, which can be regarded as an indicator of category competitiveness. Note that we regarded the apps that belong to the top 1% of the most downloaded apps as **popular apps**. The result is shown in Figure 2.

Category Ranking by Total Installs. Besides the "Games" Category, "Tools" received the most number of app installs (e.g., 25 billion till 2017) across the three years. The ranking of most categories does not vary significantly during the period. Five out of the 33 categories have received higher ranking from 2014 to 2017, for

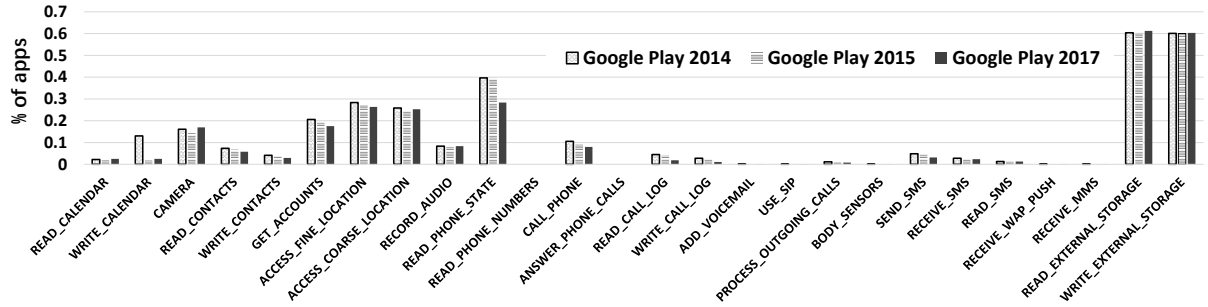


Figure 3: Evolution of requested sensitive permissions.

example, the ranking of category "Sports" went up to 11 from 21. Six categories (e.g., "Travel & Local") have received lower rankings.

Category Ranking by Average Downloads. We then compare app categories based on the average app installs, which is an indicator of app quality and user adoption. As shown in Figure 2, the general trend is stable, except for a few categories. Although category "Entertainment" receives quite high number of overall installs (rank-4 across the period), it only ranks No. 14 on average app downloads. Although the "Weather" category only ranks 19 to 22 in regard of total installs, the average app installs in this category are quite high, which ranks 5 to 8 during the evolution.

Ranking by Competitiveness. We then rank app categories based on the percentage of popular apps in each category. "Business" and "Medical" are the most competitive categories, while the total installs and average installs of these two categories rank low among the 33 categories. "Video Player" and "Communication" categories are the least competitive categories, while they have the most number of average app installs. This result could offer insights to new app developers on their developing and releasing strategies.

4.5 Permission Usage

We then investigate how sensitive permission usage by Google Play apps has evolved over time. Permission requests protect sensitive information available from a device, while Google requires developers to use sensitive permissions only when it is necessary for the functioning of the app. We seek to investigate how the permission usage in the whole ecosystem has evolved. For each app, we extract its requested permissions. Note that we only focus on the so-called *dangerous permissions* as listed by Android [8], i.e., the 26 system permissions used to guard accesses to sensitive data.

Figure 3 shows the overall result. The trend is that the number of apps requesting sensitive permissions are on the decline. More than 13% of apps requested the WRITE_CALENDER permission in 2014, while the numbers in 2015 and 2017 are only 1.7% and 2.5%, respectively. As another example, roughly 40% of apps requested the READ_PHONE_STATE permission in 2014 and 2015, while the number went down to 28% in 2017. Note that we will discuss the evolution of per-app permission request in Section 5.

One possible reason leading to this result is that Google Play requires all developers to provide a privacy policy since 2017 when an app requests sensitive permissions, which may in turn help apps respect users' privacy (i.e., eliminate unnecessary permissions).

4.6 Privacy Policies

Since 2017, Google Play requires developers to provide a valid privacy policy when an app requests or handles sensitive information [47]. It is reported that Google Play will remove apps with no privacy policies [1]. Thus, we can compare the developers' practices on privacy policy declaration along with the timeline. Note that the dataset we crawled in 2014 does not contain the privacy policy URL of the corresponding app, thus we only compare the apps in the 2015 and 2017 snapshots.

Our empirical analysis suggests that Google Play apps may declare privacy policy in either the app description or in the field of privacy policy shown on web pages. Thus, for the apps that request sensitive permissions, we check whether it declares privacy policies in its app description by (1) searching keywords "Privacy Policy" or (2) inspecting the privacy policy field shown on app pages. Note that we only consider whether the privacy policy exists, i.e., it should declare a privacy policy as long as it requests sensitive information, as required by Google Play. We did not check whether the privacy policy is consistent with its actual behaviors, which is another interesting topic that is out of the scope of this paper [73].

Figure 4 shows the analysis result for each sensitive permission. For the apps that request a specific permission, we calculate the percentage of apps with privacy policies. The results show that, in 2015, less than 20% of apps (11% on average) declared privacy policies even if they have requested dangerous permissions. For example, 216,160 apps requested the "CAMERA" permission in 2015, while only 31,814 of them (14.7%) declared privacy policies. This scenario has been greatly improved in 2017. For most permissions, more than half of the apps that used these permissions have declared privacy policies. The percentage is generally four times higher than that in 2015. For example, more than 62.8% of apps that use the "CAMERA" permission now have declared privacy policies.

Implication. This result suggests that the guidelines of Google Play on privacy policies have greatly improved the privacy protection of mobile users. Nevertheless, we should notice that a considerable number of apps still have not declared privacy policies, even if they request access to users' sensitive data, which indicates that they may violate Google Play's policy but are not removed.

4.7 Mobile Advertisement Services

Third-party libraries form an integral part of the mobile ecosystem, helping ease app development and enable features such as

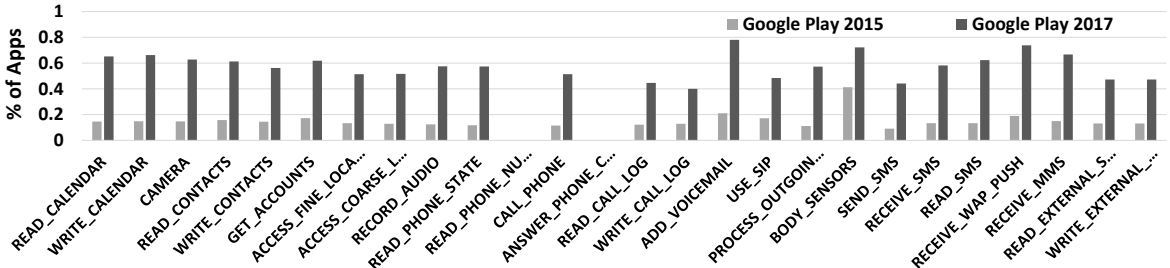


Figure 4: The percentage of privacy policy declaration for each sensitive permission.

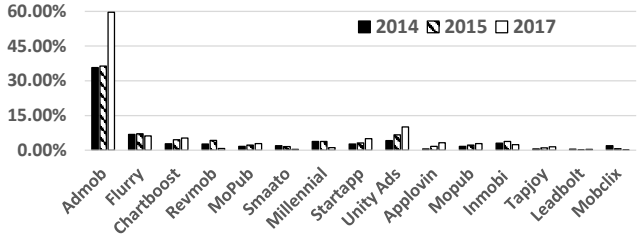


Figure 5: Top 15 mobile advertisement libraries.

analytics, social networking, and app monetization through advertisements. However, previous work suggested that third-party tracking services, especially the mobile advertisement services, may bring security and privacy issues to mobile users.

One important question here we want to explore is *whether more Google Play apps have embedded advertisement services during our investigation period.* Thus, we first need to identify third-party libraries in the apps we crawled. Here we extend a state-of-the-art clustering-based approach [43] to the millions of apps we crawled, and build a complete list of third-party libraries used in Google Play apps. The key idea is that, code that belongs to third-party libraries would be grouped in a large cluster. Then we manually labelled ad libraries based on several existing classifications, including AppBrain library classification [9], PrivacyGrade classification [6], and Common Library classification [36]. At last, we have manually labeled over 100 ad libraries.

Overall, 43.8% of apps in 2014 used at least one ad library, while over 68% of apps in 2017 embedded ad libraries. It means that more and more Google Play apps tend to use advertisement libraries, which is consistent with our findings in Section 4.1 that the app monetization schemes have changed over time, i.e., the number of paid apps is decreasing and most free apps rely on mobile advertisement to make money. Figure 5 further shows the top 15 mobile advertisement libraries used in Google Play apps. It is no doubt that Admob is dominating Google Play all the time. Over 35% of apps in the 2014 snapshot use the Admob library, while the percentage has raised to roughly 60% in our 2017 dataset.

5 LONG-LASTING APPS

In this section, we focus on the life cycle of apps. We define "**long-lasting apps**" as those available in all three Google Play snapshots, and we define "**removed**" apps as those apps that were removed from Google Play in either the 2015 or 2017 snapshot. In our dataset, 743,530 apps are flagged as "long-lasting apps", i.e., we have crawled three snapshots of them. It indicates that over 50% of the apps released in 2014 have been removed during the four-year period. We will next investigate them from different perspectives, including *app growth rate*, *app updates* and *permission evolution.*

5.1 Growth Rate

It is important to study the growth rate of app installs for the long-lasting apps, which could offer insights on how to develop successful apps. Thus, we first classify the number of app installs into 10 ranges, e.g., $[1, 10)$, $[10, 100)$, $[100, 1K)$, etc. Then, we analyze how many of them keep staying in the original ranges, and how many of them have evolved to higher install ranges.

Figure 6(a) presents the results. Overall, more than 57% of apps did not move ahead to a higher install ranges, especially for the apps with installs higher than 10K. For the apps with less than 1K downloads in the 2014 snapshot, roughly 40% of them have moved into a higher range during our study period. Interestingly, very few unpopular apps (roughly 0.04%) would become popular (with more than 1 million installs) later, as almost all of them would remain unpopular and with relative low downloads. This result suggests that the *initial status of the app would determine whether this app could be successful to a great extent.*

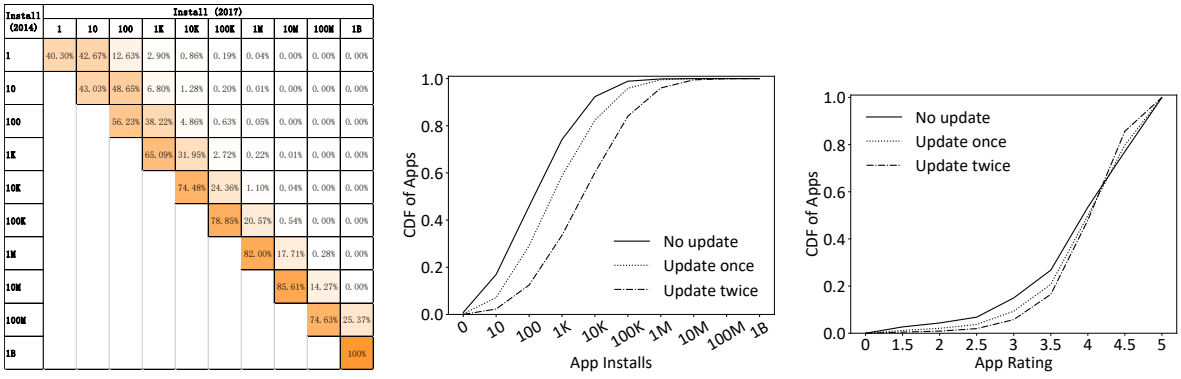
5.2 App Updates

We then study the statistics on app updates, which form a critical part of the app life cycle and would greatly affect app quality. It is important to understand how apps update over time, especially as several new Android versions was introduced during the period of our snapshots. For example, Lollipop (version 5.0) was released in October 2014, Marshmallow (version 6.0) was introduced in May 2015, and Nougat (version 7.0) was announced in August 2016.

Percentage of Updated Apps. For each long-lasting app, we have analyzed its three snapshots by comparing the hash values of the apks and the extracted version names and version code. To our surprise, **roughly 61% of apps have not received any update within our observation period**. Only 11.44% of them have been updated at least twice[2]. This means that more than 60% apps did not release new versions for almost four years, even if the Android system has evolved over several versions, and introduced several key features such as runtime permission enforcement [2].

App Updates across Categories. We then investigate whether there is a correlation between app updates and app categories, as

[2]Our sampling only covered 3 snapshots, while the apps may update more than twice.



(a) The growth rate of app installs. (b) The distribution of app installs. (c) The distribution of app rating.

Figure 6: The statistics of long-lasting apps (i.e., apps that appeared in all three snapshots).

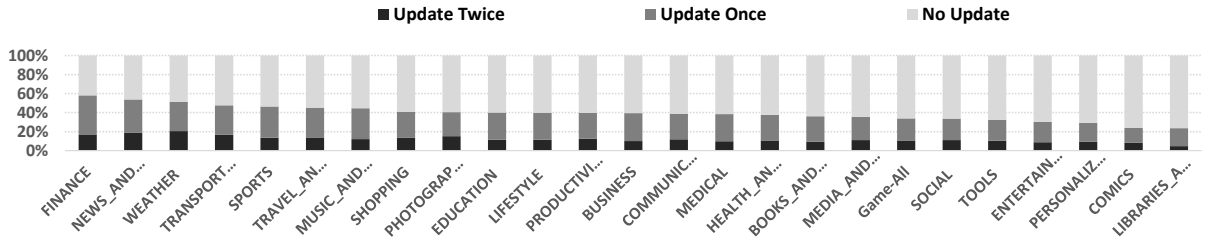


Figure 7: The distribution of app updates across categories.

shown in Figure 7. Apps in "Finance" and "News and Magazine" categories have received the highest number of updates, with roughly 60% of them have been updated at least once. In contrast, categories "Comics" and "Libraries and Demo" are the most inactive, with roughly only a quarter of apps in these categories have been updated at least once. One possible reason is that apps in categories like "Libraries and Demo" contain mostly static content that do not need to be updated, while apps in categories like "Finance" contain mostly dynamic content and they need to be updated frequently in order to provide new features. Nevertheless, the percentage of apps that received frequent updates is relatively low.

App Updates vs. App Popularity We then investigate whether popular apps received more updates than the unpopular ones, or whether app updates have positive effects on app installs. Figure 6(b) shows the distribution of app installs at different "update frequency levels". It is obvious that there is a positive correlation between app updates and app popularity. Over 40% of apps in the "update twice" group have received more than 10K installs, while the percentage in the "no update" group is only 7%. We further analyze the user ratings of long-lasting apps, in order to investigate whether users are more satisfied with apps that received more updates. As shown in Figure 6(c), there is no strong correlation between app rating and app update, although the distribution of app ratings for apps updated twice is slightly better.

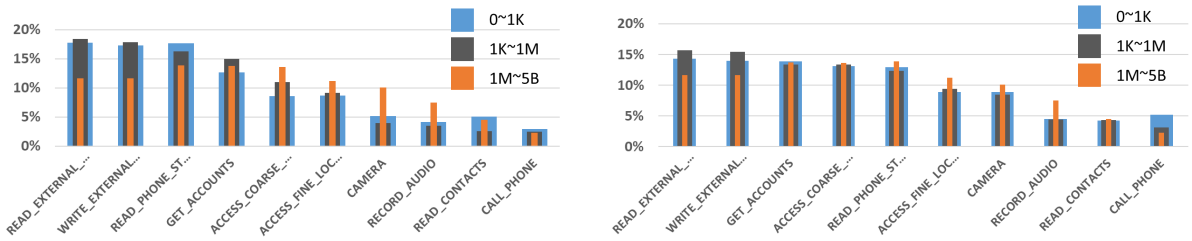
5.3 Permission Evolution

We also analyzed permission changes at the per-app level to understand how many permissions each individual app had added or removed during the evolution. For a given app, its permission usage between snapshots falls into the four categories: (1) new permissions, (2) removed permissions, (3) both new and removed permissions, (4) permissions unchanged. Between the three snapshots we created, there are two intervals where we could study permission evolution.

Table 3: The distribution of permission evolution patterns.

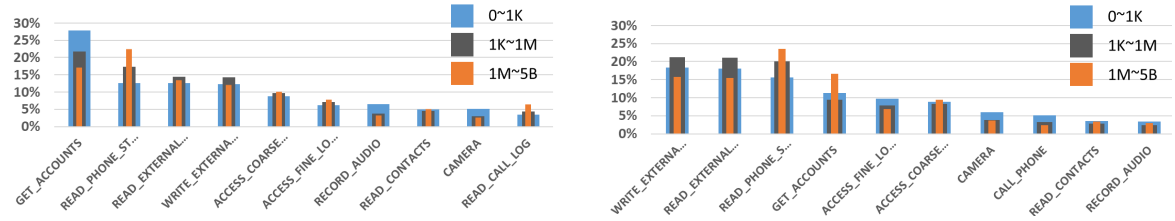
		15-17			
		new	removed	both	unchanged
14-15	new	1.03%	1.76%	1.50%	1.70%
	removed	4.12%	1.17%	1.66%	0.97%
	both	0.50%	0.53%	1.16%	0.36%
	unchanged	21.21%	38.95%	20.49%	2.90%

Changes in Permission Usage. We first investigate how many long-lasting apps changed their permission usage during the evolution. Note that our previous observation suggests that over 60% of apps did not received any updates, thus we eliminate these apps when studying the changes in permission usage. As shown in the Table 3, each cell has two corresponding values, which represent the permission evolution states from 2014 to 2015 and from 2015 to 2017, respectively. Roughly 2.9% of the apps have no permission changes during the period, while only 0.4% of apps have requested more permissions all the time. Note that over 80% of apps do not have permission changes from 2014 to 2015, while roughly 95% of apps changed their permissions from 2015 to 2017. Over 42% of apps removed sensitive permissions and did not introduce new permissions from the year 2015 to 2017.



(1) from 2014 to 2015 (2) from 2015 to 2017

Figure 8: Added permissions.



(1) from 2014 to 2015 (2) from 2015 to 2017

Figure 9: Removed permissions.

Changed Permissions. We then analyze the permissions that were added to apps when they were updated, to understand the potential privacy risks. Figure 8 shows the breakdown of the most commonly added permissions for apps with different download levels. The top 2 permissions added are "READ_EXTERNAL_ STORAGE" and "WRITE_EXTERNAL_STORAGE", especially for the apps with low installs. Many apps added the "READ_EXTERNAL_STORAGE" permission while they have already declared the corresponding write permission. What is apparently unclear to these developers is that, apps granted with the write permission have implicitly obtained the read access as well, so there is no need to ask for the read permission. Besides, popular apps tend to request more sensitive permissions such as "CAMERA" and "RECORD_AUDIO" during the evolution than the apps with lower app installs. Figure 9 shows which permissions were removed the most. "GET_ACCOUNTS" is the most commonly removed permission from 2014 to 2015, and the external storage permissions are the most frequently added and removed permissions from 2015 to 2017.

6 THE EVOLUTION OF SECURITY RISKS

In this section, we study the evolution of security risks and developing misbehaviors, i.e., the presence of malicious apps, and the spamming apps using ranking fraud techniques.

6.1 Malicious Apps

As a measure to eliminate malicious apps, Google Play has adopted some vetting process to keep malware from entering the market, e.g., the Bouncer service [52] and machine-learning based approaches [19]. However, it is reported that malicious apps are recurrently found in Google Play [4, 5, 21, 61], especially for new malware variants [3]. As Google Play is removing malware from time to time, we want to investigate whether Google Play is moving towards higher security levels during its evolution.

Table 4: Distribution of potential malware over time.

Year	AV-rank (# apps, % apps)		
	≥ 1	≥ 10	≥ 20
Google Play 2014	261,480 (17.17%)	44,664 (2.93%)	9,324 (0.61%)
Google Play 2015	250,484 (15.17%)	37,111 (2.25%)	8,716 (0.53%)
Google Play 2017	179,184 (8.35%)	19,459 (0.91%)	3,854 (0.18%)

To this end, we upload all the apps we collected from the three snapshots to VirusTotal, a popular online malware analysis service that aggregates more than 60 anti-virus engines. Note that, some anti-virus engines on VirusTotal may not always report reliable results [10, 71]. To reduce false positives, we use "AV-Score" to represent how many engines flag an app as malware. We empirically select the threshold of "$AV - Score = 10$" as a robust choice for identifying malware, as suggested by previous studies [10, 70].

Distribution of Malicious Apps We first analyze the presence of malicious apps in the three snapshots, as reported in Table 4. We are glad to report that *the proportion of potentially malicious apps in Google Play is decreasing year by year.* Roughly 50% of the apps in snapshot 2014 are flagged at least by one anti-virus engine, while the percentage for snapshot 2017 is significantly lower (8.35%). According to the threshold of "$AV - Score = 10$", around 0.91% of the apps in snapshot 2017 are labeled as malware, while the percentage in snapshot 2014 is three times higher.

Evolution of Malware Families We also analyzed the malware families that are prevalent on Google Play. We resort to AVClass [51], a widely used malware labeling tool to obtain the family name of each identified malware, and Figure 10 shows the results. The top malware families are relatively stable across the three years. "revmob" and "airpush" are the most popular malware families, which are well-known aggressive adware.

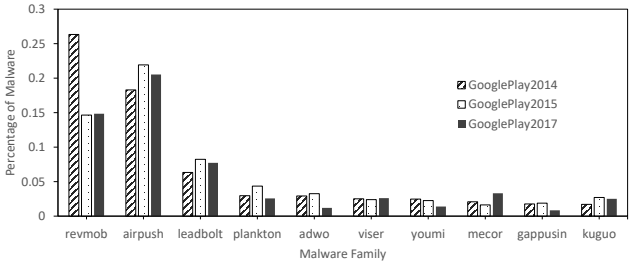


Figure 10: The distribution of the top 10 malware families.

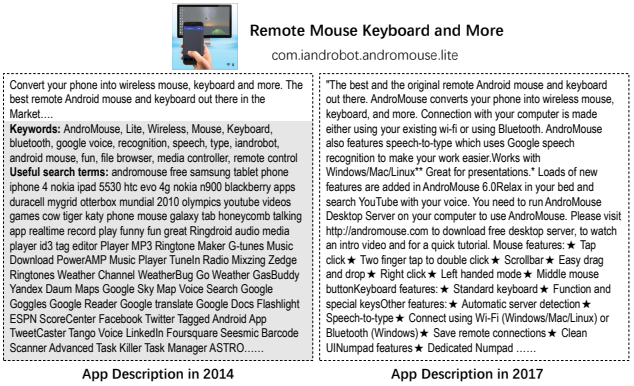


Figure 11: An example of misleading references to popular apps (left). The description has been improved later (right).

6.2 Spamming Apps

Google Play has updated its developer policy on metadata to make app listing accurate and relevant in October 2016 [24]. Google asked the developers to avoid user testimonials, excessive details, misleading references to other apps and repetitive, excessive or irrelevant keywords. The motivation is that some developers would like to insert irrelevant keywords (e.g., the names of popular apps and popular searching words) in their app descriptions, so that their apps would appear popular (highly ranked) in the search results, which is a common spamming technique and even used by some app store optimization (ASO) providers.

Approach. Therefore, it is important to investigate whether the situation of spamming apps has been improved with the evolution. To this end, we resort to the most straightforward strategy to identify spamming apps with irrelevant descriptions, which is to identify apps that insert names of popular apps in their descriptions. This strategy proposed by Seneviratne *et al.* [54] is of interest to spamming developers. As a result, we first collect the top 100 popular app names, then we count the number of popular app names mentioned in the descriptions of all the apps we harvested. Note that popular app names could be common words, such as "Video Player", "Weather" and "Music", thus we also excluded 10 such common words from this study.

Result. Table 5 summarizes the result. In general, during the evolution, more apps embed these popular app names in their descriptions. Since mobile apps may communicate with Facebook and Twitter to provide social networking functionalities, having those keywords in the app descriptions does not directly mean that these apps are spamming apps. By manually inspecting some cases, we believe that incorporating more than 10 popular app names in the description are mostly spamming apps.

The result suggests that the 2017 snapshot contains more possible spamming apps, while the aggressive level is not as serious as that in snapshot 2014 and 2015. For example, there are 56 apps in snapshot 2015 that have listed more than 40 keywords in their app descriptions (the most aggressive one has 80 keywords), while the most aggressive one in 2017 only has 25 keywords. We further explore the evolution of these spamming app candidates. About 62% of these apps identified in 2014 and 2015 snapshots were removed in 2017, and for the remaining apps, all of their app descriptions have been improved. For example, as shown in Figure 11, the app "com.iandrobot.andromouse.lite" has inserted more than 200 search keywords (named "Useful search terms") related to other popular

Table 5: The distribution of the number of misleading references to other popular apps.

Year	References (# apps)				
	≥ 1	≥ 5	≥ 10	≥ 15	≥ 20
Google Play 2014	489,795	18,479	593	122	68
Google Play 2015	792,130	39,827	1,232	261	96
Google Play 2017	1,261,613	86,730	3,013	148	20

apps in the description we crawled in 2015 (left), while the description has been improved as in 2017 (right). Nevertheless, there are still over 3,000 possible spamming apps in the 2017 snapshot. We manually examined some of them, and found that they tend to use some sophistic methods to insert keywords. Rather than listing keywords without logic, they seek to embed the keywords into real sentences. For example, "com.aditya.hexawhite" is an icon replacement app, which has incorporated lots of popular app names and keywords in its description using statements such as "compatible with: [A list of App Names]" and "Icon List: [A list of App Names]".

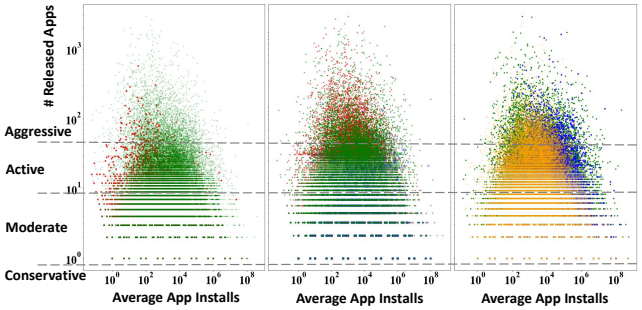


Figure 12: The distribution of app developers in Google Play over time. Green, Blue and Yellow dots represent new developers in 2014, 2015 and 2017 respectively. Red dot represents the developers removed in the next snapshot.

7 THE EVOLUTION OF APP DEVELOPERS

This section investigates the dynamics of app developers. Table 2 shows that there are roughly 400K to 500K developers in Google

Table 6: Developer groups.

Snapshot	Category (# developers, % Percentage)			
	Conservative	Moderate	Active	Aggressive
GP2014	239,331 (57.8%)	151,541 (36.6%)	20,219 (4.9%)	2,869 (0.7%)
GP2015	211,029 (52.9%)	156,961 (39.4%)	27,737 (7.0%)	3,128 (0.8%)
GP2017	306,996 (56.7%)	194,544 (36.0%)	35,066 (6.5%)	4,499 (0.8%)

Play during the span of our study. However, as we plotted in Figure 12, developers change greatly during the evolution, as a considerable number of developers have disappeared, while new developers have emerged. Only 42% of developers in the 2017 snapshot are originated from 2014. Next, we first study the developer popularity, then we classify developer into different groups, and further characterize spamming and malicious developers.

7.1 Top Developers

We characterize the developer popularity distribution for all the developers ordered by the number of accumulated installs, as illustrated in Figure 13(a). This result suggests that the distribution of developer installs also follows the Pareto effect. Top 1% of developers have occupied roughly 83% of total installs across the three snapshots. We further compare the catalog similarity of the top 1% of developers, and found that more than 80% of them are the same. Thus, although the active developers change greatly across snapshots, the most popular ones are relative stable, so that Google Play is dominated by a small number of developers all the time.

7.2 Developer Categorization

Previous work [69] proposed to classify developers into four groups based on the number of apps they released, including Conservative (1 app), Moderate (2 to 9 apps), Active (10 to 49 apps) and Aggressive (more than 50 apps). In this work, we follow this classification to study the evolution of app developers.

Table 6 and Figure 12 shows the distribution of different developer groups. The developer distribution follows the similar pattern. Conservative developers account for more than half of the community, while less than 1% of the developers are aggressive developers.

Spamming Developers. Previous work identified the prevalence of spamming developers in Google Play, who have released a large number of low-quality apps (mostly are clones) with almost no downloads. This study tries to study the evolution of spamming developers, i.e., whether spamming developers left or are removed. Following the same criteria (aggressive developers with no popular apps and with an average install number lower than 10,000), we flag each developer as spamming or not. We have identified the number of spamming developers in these three snapshots, with numbers at 2,378, 2,474 and 3,299 respectively.

Removal of Spamming Developers. Figure 12 indicates that a large number of developers in the 2014 and 2015 snapshots are removed. We are interested in whether spamming developers were removed or not, because many of them released hundreds of low-quality and least popular apps, which should be removed. Thus we analyze the percentage of removed apps for spamming developers we found in the the 2014 and 2015 snapshots, as shown in Figure 13(b). We can see that, more than 40% of the spamming developers are removed from Google Play, with some of them having released hundreds of apps. More than 40% of the released apps have been removed for roughly 60% of the spamming developers. For example, developer "ReverbNation Artists" had released 2,634 apps in total (with average installs of 226) in the snapshot 2015, while it was removed along with all its 2,634 apps in 2017.

7.3 Malicious Developers

We are also interested in the malicious developers who have released malware to Google Play. Thus, we regard the developers who have created at least one malicious app (with AV-Score >=10) as a malicious developer. The number of malicious developers is 13,488 (3.26%), 12,894 (3.23%) and 8,621 (1.59%), respectively. The most aggressive developers have released hundreds of malware. For example, "Apps Ministry LLC" has distributed 611 malware, and "AppShareNI" released 448 malware. We further analyze whether these malicious apps are dominated by malicious developers, as shown in Figure 13(c). We can see that, top 10% malicious developers account for roughly 50% to 60% of the malicious apps.

8 DISCUSSION

We believe that our efforts and the revealed insights can contribute to different stakeholders of the mobile app ecosystem.

App Markets. For market maintainers, because Google Play may remove low-quality and policy-violation apps regularly, the lessons learned from this longitudinal study could help app markets understand real threats and challenges, and help them further improve the markets. For example, market maintainers should pay more attention to the fraudulent app promotion activities mentioned in this paper. Besides, although the percentage of malicious apps and malicious developers in Google Play are decreasing year by year, there still exist new emerging malicious apps, while aggressive developers may still be able to release a large number of malware to the market.

App Developers and App Users. From the perspective of app developers, various findings revealed in this paper could help them make better decision on how to develop and release apps, e.g., app categories, app monetization schemes, and app growth rate, etc. For mobile users, it is better to avoid apps that have not been updated for years, and avoid the potentially risk/low-quality apps that were released by malicious and spamming developers.

9 CONCLUDING REMARKS

We conduct a large-scale longitudinal study to understand the evolution of mobile app ecosystems. Specifically, our analysis covers over 5 million app records collected from three different snapshots of Google Play over a period of more than three years. Overall, our findings suggest that the overall ecosystem shows a promising progress in regard of app popularity, user ratings and privacy respecting. However, there still exists a considerable number of unsolved issues including malware, app update issues, aggressive app promotion behaviors, and spamming/malicious developers. We believe that our research efforts can positively contribute to benefiting app users and developers, attracting the focus of the research community and regulators, and advocating best operational practices across app market operators.

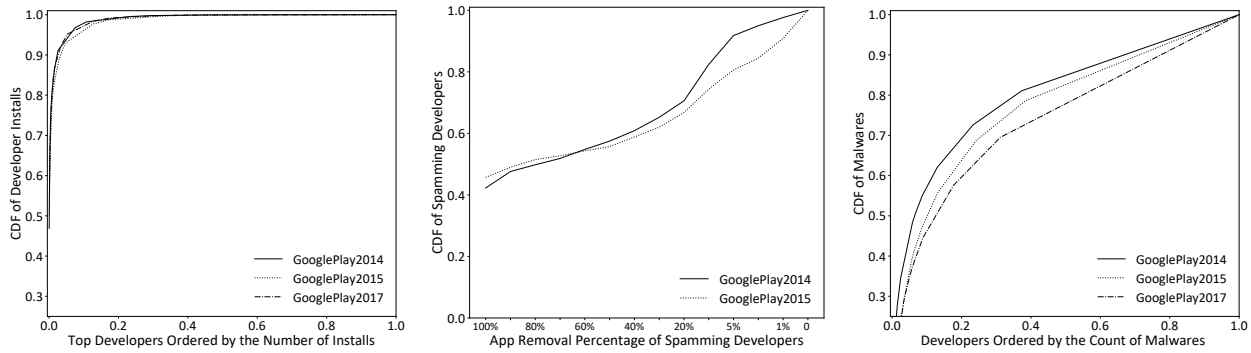


Figure 13: Statistics of app developers. (a) Top developers. (2) Removal of spamming developers. (c) Malicious developers.

ACKNOWLEDGEMENT

This work is supported by the National Key Research and Development Program of China (grant No.2017YFB0801903) and the National Natural Science Foundation of China (grants No.61702045 and No.61772042). Yao Guo is the Corresponding author.

REFERENCES

[1] 2017. Google To Delete Android Apps over Privacy Policy. https://www.techrepublic.com/article/google-will-soon-delete-apps-with-no-privacy-policies-from-play-store/.

[2] 2017. Request App Permissions. https://developer.android.com/training/permissions/requesting.

[3] 2018. Google Play is hosting a disturbing amount of cryptocurrency malware. https://thenextweb.com/hardfork/2018/04/20/google-play-cryptocurrency-apps-malware/.

[4] 2018. How Malware Keeps Sneaking Past Google Play's Defenses. https://www.wired.com/story/google-play-store-malware/.

[5] 2018. New Android Trojan malware discovered in Google Play. https://blog.malwarebytes.com/cybercrime/2017/11/new-trojan-malware-discovered-google-play/.

[6] 2018. PrivacyGrade: Grading The Privacy Of Smartphone Apps. privacy-grade.org/.

[7] Mohamed Ali, Mona Erfani Joorabchi, and Ali Mesbah. 2017. Same app, different app stores: A comparative study. In *Proceedings of the 4th International Conference on Mobile Software Engineering and Systems*. 79–90.

[8] Android. 2018. Permissions overview. https://developer.android.com/guide/topics/permissions/overview#permission-groups.

[9] AppBrain. 2018. Android Ad Network statistics and market share. https://www.appbrain.com/stats/libraries/ad-networks.

[10] Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, Konrad Rieck, and CERT Siemens. 2014. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket.. In *The Network and Distributed System Security Symposium (NDSS '14)*. 23–26.

[11] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. 2014. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. *Acm Sigplan Notices* 49, 6 (2014), 259–269.

[12] Michael Backes, Sven Bugiel, and Erik Derr. 2016. Reliable third-party library detection in android and its security applications. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 356–367.

[13] Paolo Calciati and Alessandra Gorla. 2017. How do apps evolve in their permission requests?: a preliminary study. In *Proceedings of the 14th International Conference on Mining Software Repositories*. 37–41.

[14] Bogdan Carbunar and Rahul Potharaju. 2015. A longitudinal study of the Google app market. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ACM, 242–249.

[15] Kai Chen, Peng Liu, and Yingjun Zhang. 2014. Achieving accuracy and scalability simultaneously in detecting application clones on android markets. In *Proceedings of the 36th International Conference on Software Engineering*. 175–186.

[16] Ning Chen, Jialiu Lin, Steven CH Hoi, Xiaokui Xiao, and Boshen Zhang. 2014. AR-miner: mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th International Conference on Software Engineering*. ACM, 767–778.

[17] Jonathan Crussell, Ryan Stevens, and Hao Chen. 2014. Madfraud: Investigating ad fraud in android applications. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. 123–134.

[18] Christian de Looper. 2018. From Android 1.0 to Android 9.0, here's how Google's OS evolved over a decade. https://www.digitaltrends.com/mobile/android-version-history/.

[19] Christian de Looper. 2018. Google's AI has cut down Google Play's malware by more than half. https://www.digitaltrends.com/mobile/google-play-malware-artificial-intelligence/.

[20] Feng Dong, Haoyu Wang, Li Li, Yao Guo, Tegawendé F Bissyandé, Tianming Liu, Guoai Xu, and Jacques Klein. [n. d.]. Frauddroid: Automated ad fraud detection for android apps. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 257–268.

[21] Jon Fingas. 2018. Android malware returned to Google Play with just a name change. https://www.engadget.com/2018/05/14/android-malware-returned-to-google-play-with-just-a-name-change/.

[22] Bin Fu, Jialiu Lin, Lei Li, Christos Faloutsos, Jason Hong, and Norman Sadeh. 2013. Why people hate your app: Making sense of user feedback in a mobile app store. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1276–1284.

[23] Jun Gao, Li Li, Pingfan Kong, Tegawendé F Bissyandé, and Jacques Klein. 2018. Poster: On Vulnerability Evolution in Android Apps. In *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*. 276–277.

[24] GooglePlay. 2018. Store Listing and Promotion. https://play.google.com/intl/en-US/about/storelisting-promotional/metadata/index.html.

[25] Michael Grace, Yajin Zhou, Qiang Zhang, Shihong Zou, and Xuxian Jiang. 2012. Riskranker: scalable and accurate zero-day android malware detection. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 281–294.

[26] Michael C Grace, Wu Zhou, Xuxian Jiang, and Ahmad-Reza Sadeghi. 2012. Unsafe exposure analysis of mobile in-app advertisements. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. 101–112.

[27] JERRY HILDENBRAND. 2018. 10 years of Android: How the best mobile OS was made. https://www.androidcentral.com/10-years-android-how-best-mobile-os-was-made.

[28] Yangyu Hu, Haoyu Wang, Li Li, Yao Guo, Guoai Xu, and Ren He. 2019. Want to Earn a Few Extra Bucks? A First Look at Money-making Apps. In *IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER 2019)*.

[29] Yangyu Hu, Haoyu Wang, Yajin Zhou, Yao Guo, Li Li, Bingxuan Luo, and Fangren Xu. 2018. Dating with scambots: Understanding the ecosystem of fraudulent dating applications. *arXiv preprint arXiv:1807.04901* (2018).

[30] Muhammad Ikram, Narseo Vallina-Rodriguez, Suranga Seneviratne, Mohamed Ali Kaafar, and Vern Paxson. 2016. An analysis of the privacy and security risks of android vpn permission-enabled apps. In *Proceedings of the 2016 Internet Measurement Conference*. ACM, 349–364.

[31] Yuta Ishii, Takuya Watanabe, Fumihiro Kanei, Yuta Takata, Eitaro Shioji, Mitsuaki Akiyama, Takeshi Yagi, Bo Sun, and Tatsuya Mori. 2017. Understanding the security management of global third-party Android marketplaces. In *Proceedings of the 2nd ACM SIGSOFT International Workshop on App Market Analytics*. 12–18.

[32] Mona Erfani Joorabchi, Ali Mesbah, and Philippe Kruchten. 2013. Real challenges in mobile app development. In *The ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 15–24.

[33] Hammad Khalid, Emad Shihab, Meiyappan Nagappan, and Ahmed E Hassan. 2015. What do mobile app users complain about? *IEEE Software* 32, 3 (2015), 70–77.

[34] Chengze Li, Haoyu Wang, Junfeng Wang, Qi Li, Jianbo Yu, Jingyi Guo, Guoai Xu, and Yanhui Guo. 2017. CRSPR: PageRank for Android apps. *IEEE Access* 5 (2017), 18004–18015.

[35] Li Li, Alexandre Bartel, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon. 2015. Apkcombiner: Combining multiple android apps to support inter-app analysis. In *IFIP International Information Security Conference*. Springer, 513–527.

[36] Li Li, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon. 2016. An investigation into the use of common libraries in android apps. In *The 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. 403–414.

[37] Li Li, Tegawendé F Bissyandé, Haoyu Wang, and Jacques Klein. 2018. Cid: Automating the detection of api-related compatibility issues in android apps. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*. ACM, 153–163.

[38] Li Li, Jun Gao, Médéric Hurier, Pingfan Kong, Tegawendé F Bissyandé, Alexandre Bartel, Jacques Klein, and Yves Le Traon. 2017. AndroZoo++: Collecting Millions of Android Apps and Their Metadata for the Research Community. *arXiv preprint arXiv:1709.05281* (2017).

[39] Yuanchun Li, Yao Guo, and Xiangqun Chen. 2016. Peruim: Understanding mobile application privacy with permission-ui mapping. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 682–693.

[40] Soo Ling Lim and Peter J Bentley. 2012. How to be a successful app developer: Lessons from the simulation of an app ecosystem. *ACM SIGEVOlution* 6, 1 (2012), 2–15.

[41] Jialiu Lin, Shahriyar Amini, Jason I Hong, Norman Sadeh, Janne Lindqvist, and Joy Zhang. 2012. Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 501–510.

[42] Minxing Liu, Haoyu Wang, Yao Guo, and Jason Hong. 2016. Identifying and analyzing the privacy of apps for kids. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*. 105–110.

[43] Ziang Ma, Haoyu Wang, Yao Guo, and Xiangqun Chen. 2016. LibRadar: fast and accurate detection of third-party libraries in Android apps. In *Proceedings of the 38th International Conference on Software Engineering Companion*. 653–656.

[44] Ziang Ma, Haoyu Wang, Yao Guo, and Xiangqun Chen. 2016. LibRadar: fast and accurate detection of third-party libraries in Android apps. In *Proceedings of the 38th international conference on software engineering companion*. ACM, 653–656.

[45] Sarah Perez. 2017. App economy to grow to $6.3 trillion in 2021, user base to nearly double to 6.3 billion. https://techcrunch.com/tag/app-ecosystem/.

[46] Thanasis Petsas, Antonis Papadogiannakis, Michalis Polychronakis, Evangelos P Markatos, and Thomas Karagiannis. 2017. Measurement, modeling, and analysis of the mobile app ecosystem. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* 2, 2 (2017), 7.

[47] Google Play. 2018. Privacy, Security, and Deception. https://play.google.com/about/privacy-security-deception/.

[48] Ben Popper. 2017. Google announces over 2 billion monthly active devices on Android. https://www.theverge.com/2017/5/17/15654454/android-reaches-2-billion-monthly-active-users.

[49] Rahul Potharaju, Mizanur Rahman, and Bogdan Carbunar. 2017. A Longitudinal Study of Google Play. *IEEE Transactions on computational social systems* 4, 3 (2017), 135–149.

[50] Israel J Mojica Ruiz, Meiyappan Nagappan, Bram Adams, Thorsten Berger, Steffen Dienst, and Ahmed E Hassan. 2016. Examining the rating system used in mobile-app stores. *IEEE Software* 33, 6 (2016), 86–92.

[51] Marcos Sebastián, Richard Rivera, Platon Kotzias, and Juan Caballero. 2016. Avclass: A tool for massive malware labeling. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. 230–253.

[52] TrendLabs Security. 2018. A Look at Google Bouncer. https://blog.trendmicro.com/trendlabs-security-intelligence/a-look-at-google-bouncer/.

[53] Suranga Seneviratne, Harini Kolamunna, and Aruna Seneviratne. 2015. A measurement study of tracking in paid mobile applications. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 7.

[54] Suranga Seneviratne, Aruna Seneviratne, Mohamed Ali Kaafar, Anirban Mahanti, and Prasant Mohapatra. 2015. Early detection of spam mobile apps. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. 949–959.

[55] Statista. 2018. Android - Statistics & Facts. https://www.statista.com/topics/876/android/.

[56] Vincent F Taylor and Ivan Martinovic. 2017. A longitudinal study of financial apps in the Google Play Store. In *2017 International Conference on Financial Cryptography and Data Security*.

[57] Vincent F Taylor and Ivan Martinovic. 2017. To update or not to update: Insights from a two-year study of android app evolution. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 45–57.

[58] Yuan Tian, Meiyappan Nagappan, David Lo, and Ahmed E Hassan. 2015. What are the characteristics of high-rated apps? a case study on free android applications. In *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*. 301–310.

[59] Alok Tongaonkar, Shuaifu Dai, Antonio Nucci, and Dawn Song. 2013. Understanding mobile app usage patterns using in-app advertisements. In *International Conference on Passive and Active Network Measurement*. 63–72.

[60] Nicolas Viennot, Edward Garcia, and Jason Nieh. 2014. A measurement study of google play. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 42. ACM, 221–233.

[61] Jaikumar Vijayan. 2018. Google Removes 145 Malware-Laden Apps From Play Store. http://www.eweek.com/mobile/google-removes-145-malware-laden-apps-from-play-store.

[62] Haoyu Wang and Yao Guo. 2017. Understanding third-party libraries in mobile app analysis. In *Software Engineering Companion (ICSE-C)*. 515–516.

[63] Haoyu Wang, Yao Guo, Ziang Ma, and Xiangqun Chen. 2015. WuKong: a scalable and accurate two-phase approach to Android app clone detection. In *Proceedings of the 2015 International Symposium on Software Testing and Analysis*. 71–82.

[64] Haoyu Wang, Yao Guo, Ziang Ma, and Xiangqun Chen. 2017. Automated Detection and Classification of Third-Party Libraries in Large Scale Android Apps. *Journal of Software* 28, 6 (2017), 1373–1388.

[65] Haoyu Wang, Yao Guo, Zihao Tang, Guangdong Bai, and Xiangqun Chen. 2015. Reevaluating android permission gaps with static and dynamic analysis. In *2015 IEEE Global Communications Conference (GLOBECOM)*. 1–6.

[66] Haoyu Wang, Jason Hong, and Yao Guo. 2015. Using text mining to infer the purpose of permission use in mobile apps. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 1107–1118.

[67] Haoyu Wang, Hao Li, Li Li, Yao Guo, and Guoai Xu. 2018. Why are Android apps removed from Google Play?: a large-scale empirical study. In *Proceedings of the 15th International Conference on Mining Software Repositories*. ACM, 231–242.

[68] Haoyu Wang, Yuanchun Li, Yao Guo, Yuvraj Agarwal, and Jason I Hong. 2017. Understanding the purpose of permission use in mobile apps. *ACM Transactions on Information Systems (TOIS)* 35, 4 (2017), 43.

[69] Haoyu Wang, Zhe Liu, Yao Guo, Xiangqun Chen, Miao Zhang, Guoai Xu, and Jason Hong. 2017. An explorative study of the mobile app ecosystem from app developers' perspective. In *Proceedings of the 26th International Conference on World Wide Web*. 163–172.

[70] Haoyu Wang, Zhe Liu, Jingyue Liang, Narseo Vallina-Rodriguez, Yao Guo, Li Li, Juan Tapiador, Jingcun Cao, and Guoai Xu. 2018. Beyond Google Play: A Large-Scale Comparative Study of Chinese Android App Markets. In *The Internet Measurement Conference (IMC '18)*.

[71] Fengguo Wei, Yuping Li, Sankardas Roy, Xinming Ou, and Wu Zhou. 2017. Deep ground truth analysis of current android malware. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. 252–276.

[72] Wikipedia. 2018. Google Play - Wikipedia. https://en.wikipedia.org/wiki/Google_Play.

[73] Le Yu, Xiapu Luo, Xule Liu, and Tao Zhang. 2016. Can we trust the privacy policies of android apps?. In *The 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '16)*. 538–549.

[74] Hengshu Zhu, Hui Xiong, Yong Ge, and Enhong Chen. 2015. Discovery of ranking fraud for mobile apps. *IEEE Transactions on knowledge and data engineering* 27, 1 (2015), 74–87.