RmvDroid: Towards A Reliable Android Malware Dataset with App Metadata

Haoyu Wang[1], Junjun Si[2], Hao Li[3], Yao Guo[4]
[1] Beijing University of Posts and Telecommunications [2] Changan Communication Technology Co., LTD.
[3] OrangeApk, Inc. [4] MOE Key Lab of HCST, Peking University

***Abstract*—A large number of research studies have been focused on detecting Android malware in recent years. As a result, a reliable and large-scale malware dataset is essential to build effective malware classifiers and evaluate the performance of different detection techniques. Although several Android malware benchmarks have been widely used in our research community, these benchmarks face several major limitations. First, most of the existing datasets are outdated and cannot reflect current malware evolution trends. Second, most of them only rely on VirusTotal to label the ground truth of malware, while some anti-virus engines on VirusTotal may not always report reliable results. Third, all of them only contain the apps themselves (apks), while other important app information (e.g., app description, user rating, and app installs) is missing, which greatly limits the usage scenarios of these datasets. In this paper, we have created a reliable Android malware dataset based on Google Play's app maintenance results over several years. We first created four snapshots of Google Play in 2014, 2015, 2017 and 2018 respectively. Then we use VirusTotal to label apps with possible sensitive behaviors, and monitor these apps on Google Play to see whether Google has removed them or not. Based on this approach, we have created a malware dataset containing 9,133 samples that belong to 56 malware families with high confidence. We believe this dataset will boost a series of research studies including Android malware detection and classification, mining apps for anomalies, and app store mining, etc.**

I. Introduction

With the explosion of mobile devices and apps [1], the number of mobile malware has been growing as well. It is reported that millions of Android malware were identified every year [2], with more and more complex and sophisticated malicious payloads and evasion techniques.

The increasing threats in the mobile app ecosystem have attracted a large number of research efforts in recent years. Various kinds of malware detection techniques have been proposed, e.g., information-flow based approaches [3], [4], behavior-based approachs [5], and machine-learning based approaches [6], [7]. Besides, some related studies [8], [9] integrate app metadata (e.g., app description and privacy policy) with apps to identify outliers and suspicious malicious behaviors, while a few studies were proposed to identify specific types of mobile malware (e.g., C&C malware [10], ransomware [11] and aggressive adware [12], [13]).

To evaluate the effectiveness of malware detection, a reliable ground truth malware dataset is essential. Although previous work reported promising results on malware detection [14], [15], most of them rely on a small and outdated Android malware dataset, which unfortunately cannot reflect the malware trends in the fast evolving mobile app ecosystem and faces several limitations.

TABLE I: The most widely used Android malware dataset.

Dataset	Time	# Malware	Method/Source	Metadata
MalGenome [16]	2010-2012	1,234	Security Reports	NO
Drebin [6]	2013	5,560	VT (2 of 10 engines)	NO
Piggybacking [17]	2016	1,136	VT (>=1 engine)	NO
AMD [18]	2010-2016	24,553	VT (>=28 engines)	NO

Widely Used Android Malware Datasets. We summarized 4 representative Android malware benchmarks, as listed in Table I. These benchmarks are widely used by the research community to evaluate the effectiveness of malware detection and malware classification approaches. However, they face the following major limitations:

- **Size and coverage.** Besides the AMD dataset [18], all the remaining datasets are small and outdated. For example, MalGenome [16] and Drebin [6] are two mostly popular datasets, which were created five years ago and contain only a limited number of samples. It is also reported that Drebin dataset has the duplication issue [19]. The AMD dataset was created in 2016, with a large number of malware samples. It contains a considerable number of samples overlapped with MalGenome and Drebin projects, as it collected samples from multiple sources including existing malware datasets.
- **Methods used to flag the ground truth.** Besides the MalGenome dataset, all the remaining three datasets rely on VirusTotal[1] to label the ground truth. It is interesting to see that, they use different thresholds of detection engines on VirusTotal to label malware samples. For example, Drebin was created based on the results of 10 famous engines on VirusTotal, i.e., one sample is selected as long as *two of the 10 engines* flagged the sample as malicious. The Piggybacking dataset used 1 engine as threshold, and AMD used 28 engines (over 50% of the engines) as threshold. Although VirusTotal is widely adopted by both academia and industry, relying only on the result of VirusTotal is not always reliable. On one hand, the detection result on VirusTotal is volatile and may change with time. For example, according to a recent study [20], roughly half of recent samples cannot be recognized by any anti-virus engines on VirusTotal by the first time of uploading. It is easy for malicious

[1]https://www.virustotal.com

2574-3864/19/$31.00 ©2019 IEEE
DOI 10.1109/MSR.2019.00067

developers to bypass these engines [21], as anti-virus vendors only deployed light-weighted engines (most of them are signature-based engines) on VirusTotal in order to achieve instant detection. On the other hand, many samples flagged by VirusTotal as malicious are not always true. For example, app "com.ponphy.engineermode" was flagged by 8 to 30 engines across its different versions[2], however this app has been listed on Google Play for over 4 years, which should not be regarded as a malicious app. Besides, there are some anti-virus engine test apps (e.g., com.androidantivirus.testvirus) on markets, which are not malware either.

- *App Metadata.* To the best of our knowledge, no previous studies have collected the metadata (e.g., app description, app ratings, etc.) related to malware samples. As a number of previous studies [8], [22] proposed to incorporate app metadata for malicious/anomaly detection, it is important to create a malware dataset with all the app metadata to facilitate malware detection evaluation.

Key idea. We propose to create a reliable Android malware dataset with high confidence, along with all the app metadata. To overcome the limitations mentioned above, we make efforts to crawl apps with all the meta information from Google Play, and we resort to the app maintenance behaviors of Google Play to help label malware. It is reported that malicious apps were recurrently found in Google Play [23], [24], while Google Play has adopted strict vetting process and taken actions to remove malware from time to time [25], [26]. Thus our key idea is: **for suspicious apps with sensitive behaviors (e.g., flagged by VirusTotal), if they were removed by Google Play during our monitoring period, we will regard them as malicious apps**. For apps flagged by VirusTotal, if they are still residing at Google Play for a long time, we will regard them as benign apps, even if they were reported by some engines as malware.

This work. To this end, we have collected four snapshots of Google Play, which were crawled in March 2014, March 2015, September 2017, and November 2018 respectively (cf. **Section II**). We investigated the number of malicious apps flagged by VirusTotal, and then we checked whether any of them were removed during our following Google Play snapshots. We created a list of 10,439 malware samples, which were flagged by VirusTotal by at least 20 engines (by the time of our study) and also removed by Google Play in the snapshots we collected. We further use AVClass [27] to label the malware family name for each of the sample, and eliminate the samples with no family names or families with fewer than 5 samples. At last, we created *RmvDroid*, a dataset with 9,133 malware samples that belong to 56 malware families with all the apk files and metadata (cf. **Section III**).

To the best of our knowledge, RmvDroid is the first large-scale and *reliable* Android malware dataset along with all their *meta information*. We believe our dataset could boost the research studies including malware detection and classification, malicious developer identification, outlier detection, etc. This dataset could be accessed at:

https://zenodo.org/record/2593596

[2]Its most recent version was flagged by 11 anti-virus engines, https://www.virustotal.com//file/ec1b7b47727427dc277372b436c0bee12cdc02e161a363dd2f2ba4572fd3dda9/detection

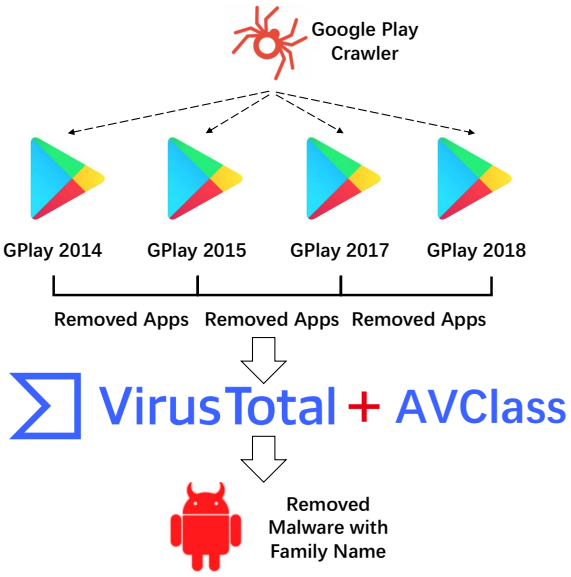

Fig. 1: The process to create the RmvDroid malware dataset.

II. Data Collection

In this Section, we introduce the data collection process, which is shown in Figure 1. We first created a Google Play crawler to index apps listed in Google Play, and download the app metadata (e.g., app names, app descriptions, developer names, user ratings, the number of app installs, etc.) and apks through Google Play API [28]. Taking advantage of this crawler, we have created four snapshots of Google Play, which were crawled in 2014, 2015, 2017 and 2018, respectively. For each snapshot we collected, we further identify how many apps have been removed by comparing these snapshots. Then, we uploaded all the apps to VirusTotal to check how many of them have been flagged by anti-virus engines and then further used AVClass [27] to assign them malware family labels.

A. Creating the Snapshots of Google Play

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.

The *first Snapshot* 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) the "Similar Apps" section shown on the app web pages recommended by Google Play and (2) other apps released by the same developer. We have crawled over 1.5 million apps, which represents almost all the apps that can be crawled from Google Play at that time. The *Second Snapshot* was created in March 2015 (1 year after the first snapshot), we repeated the same process as described above

TABLE II: Overview of the RmvDroid dataset.

	# Removed Malware (VT>=20)	# Removed Malware (filtered)
GPlay 2014	2,309	2,207
GPlay 2015	5,485	4,786
GPlay 2017	2,645	2,140
Total	10,439	9,133

to crawl Google Play, except that we take the previous 1.5 million crawled apps as our searching seeds. Overall, we are able to collect over 1.6 million Android apps in this snapshot. We repeated the same process in September 2017 (2.5 years after the second snapshot) to create the *Third 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. For the *Fourth Snapshot*, we only checked the 2.1 million apps crawled in the third Snapshot (2017) to see whether the crawled apps were removed by Google.

B. Identifying Removed Apps

Google Play is constantly removing apps according to its developers' policies[3]. To identify the removed apps, for the adjacent two snapshots (e.g., 2014 and 2015), we pinpoint the apps belonging to the first snapshot but do not exist in the second snapshot. The retained apps can then be safely considered as removed apps. In this way, we could create a list of removed apps in Google Play 2014 (removed in 2015), 2015 (removed in 2017) and 2017 (removed in 2018) separately.

C. Removed Malicious Apps

Here, we use VirusTotal to flag the sensitive behaviors of all the apps we collected. Note that, although VirusTotal is not always reliable as we mentioned above, our hypothesis here is that if the sample is flagged by VirusTotal and further removed by Google Play, it is highly possible to be confirmed as malware. For the identified malware, we further use AVClass [27], a widely used malware labelling tool to get their malware family names. Note that if no family names were found, AVClass would label the malware as "Singleton". Thus, we further eliminate the apps with no family names and families with fewer than 5 samples.

III. Overview of RmvDroid

A. Dataset Overview

The basic statistics of RmvDroid are listed in Table II. We have identified over 10K apps that have been flagged by more than 20 anti-virus engines as malware and further removed by Google Play. These apps belong to 175 malware families in total. After filtering the apps with no family name and families with samples fewer than 5, we have obtained 9,133 malware samples, which belong to 56 malware families.

As shown in Figure 2, besides the malicious apks, for each malware sample, we also collected additional information including app name, package name, app installs, user rating, cateogry, developer name, app description, the number of flagged engines on VirusTotal, and the malware family name.

[3]https://play.google.com/about/developer-content-policy/

Fig. 2: Example of one sample in the RmvDroid dataset.

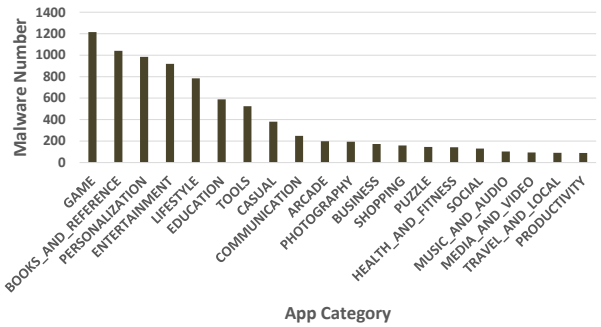

Fig. 3: The distribution of malware samples across categories.

B. Malware Distribution

Distribution Across Categories. These malware samples were found in all the categories at Google Play. Figure 3 shows the top 20 app categories that host the most number of malware. Malicious apps were found most in the GAME category, BOOKS and REFERENCE and PERSONALIZATION categories. This data could be further investigated by correlating with malware families.

Distribution of App Installs. As shown in Figure 4 (1), although over 80% of malicious apps have aggregated app installs less than 10K, there are 10 apps with installs higher than 10 million. This data could be further investigated to study the real-world impact of malware.

Distribution of App Ratings We further characterize the app ratings of our dataset, as shown in Figure 4 (2). It is interesting to see that, only less than 20% of them have ratings smaller than 1, over 40% of them have achieved high ratings (>=4). As these samples are malicious apps and removed by Google Play, it is interesting to further explore why they achieved such high user ratings. One possible reason might be that malicious developers may use fraudulent app promotion techniques [29]–[31] so the app ratings are unreliable.

C. The Distribution of Malware Families

These 9,133 samples belong to 56 different malware families. We further analyze the distribution of the top 15 malware families, as shown in Figure 5. The Airpush family accounts

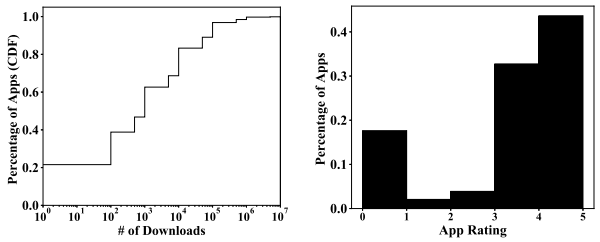

Fig. 4: The distribution of (1) app downloads (left) and (2) app ratings (right) of the malware samples.

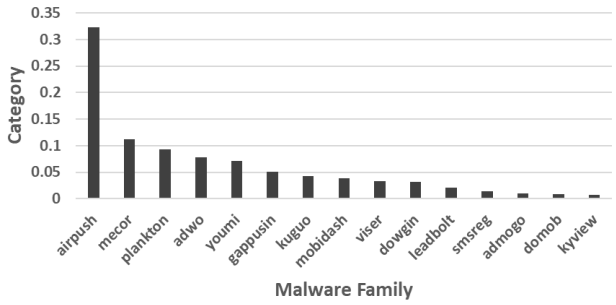

Fig. 5: The distribution of the top 15 malware families in RmvDroid dataset.

for over 30% of the samples, while the top 3 families take up over 50% of the samples.

IV. Usage Scenarios

We believe our dataset could help boost the following research areas.

Malware Detection and Classification. As we mentioned earlier, a large number of mobile malware detection and classification techniques mainly use outdated and small malware benchmarks for evaluation. The released dataset in this paper could be further leveraged to verify the validity of new/existing malware detection methods. Furthermore, as our dataset has collected samples that belong to 56 families, malware classification work could also benefit from our dataset.

Checking App Behavior against Meta Information. Some related studies [8], [9] integrate app metadata (e.g., app description and privacy policy) with apps to identify outliers. For example, CHABADA [8] was proposed to check whether an app behaves as advertised, i.e., by comparing app behaviors with app description clustering results. In their evaluation, they rely on the MalGenome project and manually search app descriptions in Google, which is not scalable and inaccurate. The dataset we released could further boost this line of studies.

Malicious Developer Analysis. This dataset could be further leveraged to study the malicious developers. For the 9,133 samples, over 3,000 developers contribute to them. Several developers are even related to hundreds of malicious apps. For example, the developer "Apps Ministry LLC" has released 471 malware, and the developer "AppShareNI" has created 234 samples. Further study could investigate the characteristics of these malicious developers, for example, analyzing the code reuse patterns and malware evolution behaviors.

Malware Impact Analysis Studying the impact caused by malware is as important as malware detection. It is not feasible to study malware impact based on the previous released benchmarks, as they do not provide the source information and app metadata. Our dataset offers this opportunity to study the impact of mobile malware.

App Market Comparison The samples in RmvDroid dataset were flagged by VirusTotal and removed by Google Play, which we believe are malicious apps with high confidence. One future research direction might be using this dataset to check their existence on other alternative markets, which could be used to enforce cross-market comparison or improve the app maintenance behaviors across app markets.

V. Limitation and Challenges

Our dataset faces several limitations. First, the malicious apps we collected may be removed by Google Play at any time during the interval between two snapshots. The metadata we collected (e.g., description, downloads, ratings, privacy policy, etc.) and the apks may change during that time. Thus, the removed malicious apps collected in this paper may not be fully representative to the situation when they were removed. Second, our hypothesis is that Google Play removed the apps with malicious behaviors, while Google Play could remove apps with a number of other reasons, which may not be triggered by the malicious behaviors. Third, we set the threshold of 20 to flag suspicious apps based on the number of reported engines on VirusTotal, while the threshold might be too high or too low. Although the threshold is configurable, it is non-trivial for us to choose the best one. At last, we still rely on the detection results of VirusTotal to label the possible malicious behaviors in removed apps, while the detection results may still volatile and change over time.

VI. Conclusion

In this paper, we present a reliable Android malware dataset collected based on four snapshots of Google Play. To overcome the challenge of malware sample labelling, we rely on both VirusTotal and Google Play's app maintenance practice. For the apps flagged by over 20 engines on VirusTotal and further be removed from Google, we will regard them as malware. As the result, we have created RmvDroid, a dataset containing 9,133 malware samples with high confidence. Our dataset could be used to facilitate a series of research studies, including malware detection and classification, mining apps for anomalies, malicious developer analysis, etc.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (grants No.61702045 and No.61772042). Yao Guo is the corresponding author.

REFERENCES

[1] H. Wang, H. Li, and Y. Guo, "Understanding the evolution of mobile app ecosystems: A longitudinal measurement study of google play," in *Proceedings of the Web Conference 2019 (WWW '19)*, 2019.

[2] "2018 Malware Forecast: the onward march of Android malware," 2017, https://nakedsecurity.sophos.com/2017/11/07/2018-malware-forecast-the-onward-march-of-android-malware/.

[3] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, p. 5, 2014.

[4] Y. Feng, S. Anand, I. Dillig, and A. Aiken, "Apposcopy: Semantics-based detection of android malware through static analysis," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, 2014, pp. 576–587.

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

[6] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket." in *Proceedings of the Network and Distributed System Security Symposium (NDSS '14)*, vol. 14, 2014, pp. 23–26.

[7] N. McLaughlin, J. Martinez del Rincon, B. Kang, S. Yerima, P. Miller, S. Sezer, Y. Safaei, E. Trickel, Z. Zhao, A. Doupe *et al.*, "Deep android malware detection," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*. ACM, 2017, pp. 301–308.

[8] A. Gorla, I. Tavecchia, F. Gross, and A. Zeller, "Checking app behavior against app descriptions," in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 1025–1035.

[9] L. Yu, X. Luo, C. Qian, S. Wang, and H. K. Leung, "Enhancing the description-to-behavior fidelity in android apps with privacy policy," *IEEE Transactions on Software Engineering*, vol. 44, no. 9, pp. 834–854, 2018.

[10] W. Yang, M. Prasad, and T. Xie, "Enmobile: Entity-based characterization and analysis of mobile malware," in *Proceedings of the 40th International Conference on Software Engineering (ICSE '18)*, 2018, pp. 384–394.

[11] J. Chen, C. Wang, Z. Zhao, K. Chen, R. Du, and G.-J. Ahn, "Uncovering the face of android ransomware: Characterization and real-time detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1286–1300, 2018.

[12] F. Dong, H. Wang, L. Li, Y. Guo, T. F. Bissyandé, T. Liu, G. Xu, and J. Klein, "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*. ACM, 2018, pp. 257–268.

[13] F. Dong, H. Wang, L. Li, Y. Guo, G. Xu, and S. Zhang, "How do mobile apps violate the behavioral policy of advertisement libraries?" in *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*. ACM, 2018, pp. 75–80.

[14] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, and M. Rajarajan, "Android security: a survey of issues, malware penetration, and defenses," *IEEE communications surveys & tutorials*, vol. 17, no. 2, pp. 998–1022, 2015.

[15] S. Arshad, M. A. Shah, A. Khan, and M. Ahmed, "Android malware detection & protection: a survey," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, pp. 463–475, 2016.

[16] X. Jiang and Y. Zhou, "Dissecting android malware: Characterization and evolution," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 95–109.

[17] L. Li, D. Li, T. F. Bissyandé, J. Klein, Y. Le Traon, D. Lo, and L. Cavallaro, "Understanding android app piggybacking: A systematic study of malicious code grafting," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1269–1284, 2017.

[18] F. Wei, Y. Li, S. Roy, X. Ou, and W. Zhou, "Deep ground truth analysis of current android malware," in *Proceedings of the 2017 International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2017, pp. 252–276.

[19] P. Irolla and A. Dey, "The duplication issue within the drebin dataset," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 3, pp. 245–249, 2018.

[20] J.-F. Lalande, V. V. T. Tong, M. Leslous, and P. Graux, "Challenges for reliable and large scale evaluation of android malware analysis," in *Proceedings of the 2018 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE, 2018, pp. 1068–1070.

[21] "Tool for leaking and bypassing Android malware detection system," 2019, https://github.com/sslab-gatech/avpass.

[22] S. Ma, S. Wang, D. Lo, R. H. Deng, and C. Sun, "Active semi-supervised approach for checking app behavior against its description," in *Proceedings of the 39th Annual Computer Software and Applications Conference*, vol. 2. IEEE, 2015, pp. 179–184.

[23] "Massive Android Malware Outbreak Invades Google Play Store," 2017, http://fortune.com/2017/09/14/google-play-android-malware/.

[24] "Malicious Android apps sneak malware onto your phone with droppers," 2017, https://mashable.com/article/droppers-malware-android-google-play-store/.

[25] H. Wang, Z. Liu, J. Liang, N. Vallina-Rodriguez, Y. Guo, L. Li, J. Tapiador, J. Cao, and G. Xu, "Beyond google play: A large-scale comparative study of chinese android app markets," in *Proceedings of the 2018 Internet Measurement Conference (IMC '18)*. ACM, 2018, pp. 293–307.

[26] H. Wang, H. Li, L. Li, Y. Guo, and G. Xu, "Why are android apps removed from google play?: A large-scale empirical study," in *Proceedings of the 15th International Conference on Mining Software Repositories (MSR '18)*. ACM, 2018, pp. 231–242.

[27] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "Avclass: A tool for massive malware labeling," in *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses (RAID '16)*. Springer, 2016, pp. 230–253.

[28] "Google Play API," 2019, https://github.com/facundoolano/google-play-api.

[29] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Discovery of ranking fraud for mobile apps," *IEEE Transactions on knowledge and data engineering*, vol. 27, no. 1, pp. 74–87, 2015.

[30] Y. Hu, H. Wang, L. Li, Y. Guo, G. Xu, and R. He, "Want to earn a few extra bucks? a first look at money-making apps," in *Proceedings of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER '19)*, 2019.

[31] Y. Hu, H. Wang, Y. Zhou, Y. Guo, L. Li, B. Luo, and F. Xu, "Dating with scambots: Understanding the ecosystem of fraudulent dating applications," *arXiv preprint arXiv:1807.04901*, 2018.