Should You Consider Adware as Malware in Your Study?

Jun Gao*, Li Li†, Pingfan Kong* Tegawendé F. Bissyandé*, Jacques Klein*
*University of Luxembourg, Luxembourg
†Monash University, Australia
{jun.gao, pingfan.kong, tegawende.bissyande, jacques.klein}@uni.lu
li.li@monash.edu

***Abstract*—Empirical validations of research approaches eventually require a curated ground truth. In studies related to Android malware, such a ground truth is built by leveraging Anti-Virus (AV) scanning reports which are often provided free through online services such as VirusTotal. Unfortunately, these reports do not offer precise information for appropriately and uniquely assigning *classes* to samples in app datasets: AV engines indeed do not have a consensus on specifying information in *labels*. Furthermore, labels often mix information related to families, types, etc. In particular, the notion of "adware" is currently blurry when it comes to maliciousness. There is thus a need to thoroughly investigate cases where adware samples can actually be associated with malware (e.g., because they are tagged as adware but could be considered as malware as well).**

In this work, we present a large-scale analytical study of Android adware samples to quantify to what extent "adware should be considered as malware". Our analysis is based on the Androzoo repository of 5 million apps with associated AV labels and leverages a state-of-the-art label harmonization tool to infer the malicious type of apps before confronting it against the ad families that each adware app is associated with. We found that all adware families include samples that are actually known to implement specific malicious behavior types. Up to 50% of samples in an ad family could be flagged as malicious. Overall the study demonstrates that adware is not necessarily benign.

***Index Terms*—Android, adware, malware**

I. Introduction

Adware is commonly known as "software that automatically displays or downloads advertising material (often unwanted) when a user is online". [1]. Following this definition, the threats to user security in adware behavior is hard to validate. Indeed, advertisement (or ad in short) has become common, particularly in mobile apps, and now constitutes the main means for free app developers to collect revenue in compensation to their efforts. This situation has made adware a necessity in the software development ecosystem and has thus delayed the exhaustive coverage of adware by security firms for 15 years after adware debut [2]. Even until now, adware remains a controversial issue: our community has still not agreed on whether adware is malware or not. The recurrent question during experimental assessments is "Should adware be taken as malware?".

MalGenome [3], a well-known Android malware dataset, which is mainly built by manual efforts, does not contain any adware. Qadri et al. [4] recently reported that researchers do not generally classify adware as malware. Nevertheless, they argue that adware, which has been flagged as such, may not solely perform advertisements (i.e., adware may also perform malicious behavior). Ishii et al. [5] in their app clone study (i.e., identifying repackaged or piggybacked Android apps) also distinguish adware and malware as from two distinct categories. Finally, according to Symantec [6] and other research works [7], [8], ad libraries may not only show ads in their apps but also can leak personal data, send SMS, etc.

These contradictory considerations suggest that our community does not agree on a clear definition of adware scope. We thus argue that there is a need in our community to clearly define the relationship between adware and malware and thereby to provide a comprehensible guideline for researchers and practitioners to follow. To this end, we resort to address this challenge in this work through a quantitative analytical study on over five millions Android apps that are crawled from various app markets, including the official one named Google Play. By sending all five million apps to VirusTotal, a free service leveraging various AV products to analyze suspicious files and URLs and facilitate the quick detection of viruses, worms, trojans, and all kinds of malware, we are able to identify all the malicious apps and their AV labels, including adware. We then leverage Euphony [9], a tool that unifies multiple AV labels, to cluster flagged adware into different families. Subsequently, we rank those adware families based on their malicious rate, i.e., how many adware in a family are also labeled as malicious, and present to the community a quantitative model which can later be leveraged to advise the malicious rates of given adware (e.g., given an adware app_{ad}, within family F, the model can suggest that this adware app has $x\%$ of chance to be a malware). Finally, by considering all the investigated apps, we showcase to the community a global relationship between adware and malware.

To summarize, this paper makes the following contributions:

- We have collected AV labels from VirusTotal for over five million Android apps, for which we plan to share with the research community to boost further studies.
- We have empirically conducted a type composition analysis for 26 selected ad families to understand the malicious status of ad families. Furthermore, we have proposed a malware probability model to quantify the global relationship between adware and malware and consequently

978-1-7281-0591-8/19/$31.00 © 2019 IEEE

SANER 2019, Hangzhou, China
ERA Track

provided a web service that leverages the proposed model to advise whether a given adware should be considered as malware.

To facilitate replication study, we make available our dataset, along with the generated intermediate dataset as well as our experimental results at

https://adwarevsmalware.github.io

The remainder of this paper is organized as follows. Section II presents the necessary background information to allow readers to better understand this work. Section III details our quantitative study towards discerning malicious adware with controlled confidence. We then enumerate the threats to validity and discuss our closely related works in Section IV and Section V respectively. Finally, Section VI concludes this paper.

II. Background

We now provide necessary background information to allow readers to better understand the purpose, techniques and key concerns of this work.

First, AVs use labeling to flag detected malicious apps (including adware). As illustrated in Fig. 1, a label is a sequence of words including information such as: type, family, platform etc. Let us take the app shown in Fig. 1 as an example, Emsisoft, an AV, labels the analyzed app as *Backdoor.AndroidOS.KungFu!IK*. From this label, we can pinpoint the type of the app is *Backdoor*, the family is *DroidKungFu*, the running platform is *android* etc.

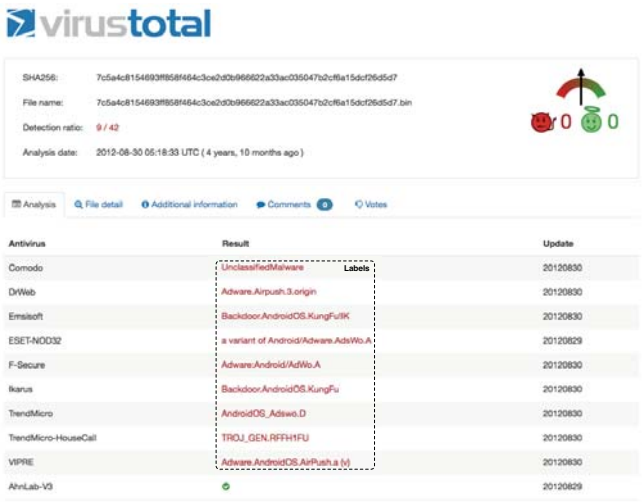

Fig. 1. An Example of VirusTotal Result.

Second, as explained, a label contains information about the type and the family of a flag malicious app. In this work, we differentiate *type* and *family* through the following definitions.

- **Type** is a behavior-oriented classification, where apps showing similar (malicious) behaviors are clustered into a same type. For example, apps showing advertisements could be categorized as adware while apps hijacking user's equipment for ransom can be taken as ransomware, etc.
- **Family** presents a more fine-grained classification where apps within one family should share some code patterns reflecting why those apps are clustered together. As examples, *kuguo* is a well-known family named after a library called kuguo. *Gingermaster* is a family where all the apps within this family target the same bug of Android platform version gingerbread.

In practice, each app will be assigned with only one type when an AV product label it (i.e., one AV product gives one type). However, we cannot blindly rely on type information to know if an adware is a "pure" adware or if this adware also performs some malicious behaviour. We can mention at least two reasons for this. First, the type given by an AV product does not necessarily mean the app's behavior will be constrained by the type definition. Indeed, it is not because an AV product gives the type adware to an app that this app cannot also perform trojan behavior. Moreover, since an AV product gives only one type, this type cannot represent all the behavior of the labeled app. The second reason relies on the fact that AV products are often inconsistent, i.e., AV products can give completely different labels leading to various possible types and families for a given app.

Therefore, in this work, we define two terms about adware to make the concept clearer:

- **Labeled adware**: Given an app scanned by certain AVs, it is tagged with an adware label by the AVs. However, whether it can implement malicious behaviours is still in doubt.
- **Pure adware**: Given an app, it is a *pure adware* when (1) it is a labeled adware, and (2) the app does not perform malicious behaviour. Hence, *pure adware* should be accepted and be taken as non-malware.

Finally, since we are going to discuss the relationships between adware and malware, in the following part, we will distinguish adware from other malware types. So hereafter, when we mention malware, adware is not included.

III. Quantitative Study

Our objective in this work is to provide a promising means for researchers and practitioners to decide whether a given adware should be considered as malware. To this end, we present the experimental setup of this work (cf. Section III-A) and two empirical studies related to adware and malware (Section III-B and Section III-C). Finally, in Section III-D, we present an implication of our approach that demonstrates the actionable usage of this work.

A. Experimental Setup

We now briefly introduce the experimental setup of this work, including the investigated Android apps, their types and families.

Android apps. The investigated apps in this work are collected from AndroZoo [10], [11], an Android application

dataset established for the research community. So far, the AndroZoo dataset contains more than five million apps crawled from 14 different markets, including the official Google Play. By running several crawlers constantly, it can provide both historical and up-to-date apps while keeping the growth of the dataset. Download APIs are also available to the research community in order to benefit other researchers by simplifying sample collection.

App Labels. All the apps in the AndroZoo dataset have been sent to VirusTotal [12], a portal to detect virus with more than 50 AV products, periodically. Thanks to this step, we have collected and maintained for every collected app a set of labels (each AV product gives one label).

App Types and Families. Labels from different AV products are usually inconsistent. For instance, in Figure 1, the label given by Emsisoft is *Backdoor.AndroidOS.KungFu!IK*, whereas the label given by F-Secure is *Adware:Android/AdWo.A*. In order to infer an unique type and an unique family for each app, we leverage a research tool called Euphony. Euphony is a state-of-the-art tool to unify AV labels in order to provide a better ground-truth to malware research studies based on labels.

B. Ad Family Study

In order to have a concrete understanding of how adware should be considered, we now investigate the malicious status of ad families. Thanks to Euphony, we have collected in total 4,088 families, including malicious and advertisement-focused ones. Based on several pre-defined ad libraries [13], by using keyword searching, we eventually identify 26 ad families in our dataset. For each family, we further conduct a type composition analysis to highlight its representative app types. The type composition analysis computes the composition rate of each type for each family. More specifically, for a family f_1, the rate is computed by considering all the apps from this family, and all their associated unique types given by Euphony. Then, we count the number of occurrences of each type. For a given type, the composition rate is finally computed as the rate between the number of occurrences of this type and the total number of occurrences of all the types. Let us consider a family with an app X of type $T1$, an app Y of type $T2$, and an app Z of type $T1$. The composition rate of $T1$ is 66.66% $(2/3)$, and the composition rate of $T2$ is 33.33% $(1/3)$.

The intuition behind this type composition analysis is that if all apps of a family have been labeled as type $adware$, the rate for the type $adware$ is 100% suggesting that apps of this family are pure adware. On the contrary, if apps of a family have been labeled with various types including adware and malware types, the rate of the type adware will be low and it could suggest that apps of this family may perform malicious behaviors.

The results of the type composition analysis are summarized in Table I, where the ad families are ranked based on their statistic malicious rate. As shown in Table I, although all these families are supposed to be ad families, they all have somehow shown malicious behavior to some extent, where **Trojan** is the most possible malware type for adware to be. Let us take family *wiyun* as an example, over 50% of investigated samples are labeled as malware. In other words, given an app labeled adware from family *wiyun*, we can quantitatively suggest that it still has 56% of chance to be malware. Similarly, for a labeled adware from family *adcolony*, the possibility of being malware of this adware is 2%.

Furthermore, considering the possibility of false alarms from AVs, a confidence rate t can be set and therefore, our study can be directly leveraged to flag labeled adware as malware, or goodware (i.e., to answer the question "Is Adware Malware?"). Given a family with a malicious rate higher than t, all adware labeled apps of this family can then be safely taken as malware, and the vice-versa.

TABLE I
AD FAMILY TYPE COMPOSITION (MALICIOUS RATE = $1 - Adware$), WHICH CAN BE TAKEN AS A LOOKUP TABLE FOR QUICKLY FLAGGING MALWARE.

Family	Labeled Adware	Labeled Malware		Malicious
		Trojan	Other Types	Rate
wiyun	44%	32%	24%	56%
feiwo	45%	40%	15%	55%
kuguo	73%	21%	6%	27%
wooboo	77%	21%	2%	23%
domob	79%	20%	1%	21%
senddroid	79%	17%	4%	21%
inmobiads	83%	14%	3%	17%
revmob	84%	15%	1%	16%
dowgin	85%	8%	7%	15%
applovin	87%	11%	2%	13%
caulyads	90%	9%	1%	10%
admobads	91%	7%	2%	9%
startapp	92%	7%	1%	8%
tapjoyads	92%	7%	1%	8%
wapsx	93%	5%	2%	7%
airpush	94%	5%	1%	6%
youmi	94%	5%	1%	6%
adwo	94%	5%	1%	6%
greystripeads	94%	5%	1%	6%
jumptapiads	95%	4%	1%	5%
adwhirlads	96%	3%	1%	4%
mobclixads	97%	2%	1%	3%
burstlyads	97%	2%	1%	3%
madhouseads	97%	2%	1%	3%
millennialmediaads	98%	1%	1%	2%
adcolony	98%	1%	1%	2%

C. Global Relationship between Adware and Malware

Based on Table I, if one of app labeled as adware was found, we can tell the probability with which it will be a malware. However, questions still remain such as "what if we don't know the family information or the family is not in the table?". In this section, we, therefore, discuss a more general method, aiming at finding the **global probability** (based on all the apps considered in this study) for a labeled adware to be a malware. Given a user-defined (or controlled) confidence, this global probability can then be leveraged to estimate an ideal *malicious rate* that further provides a promising means to flag adware as malware.

Let us now define the global probability via Formula 1.

$$P(malware|\textbf{Labeled adware}) \quad (1)$$

Labeled adware can be obtained in our sample set by checking anti-virus reported labels (or types). Since apps of one family should share the same code patterns, given a family with most of its apps labeled as adware, we should have confidence to conclude that apps of this family are likely to be **pure adware**. Based on this assumption, given an **A**dware **C**onfidence AC, all the apps of families fulfill Formula 2 should be considered as malware.

$$P(\textbf{Labeled adware}|family) < AC \quad (2)$$

Now, we can transform Formula 1 to Formula 3 shown as below (where f are the families from Formula 2):

$$\sum_{apps \in f} P(apps|\textbf{Labeled adware}) \quad (3)$$

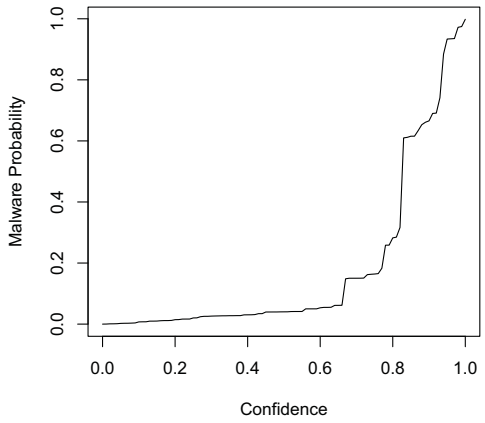

Fig. 2. Malware Probability of Labeled Adware.

Figure 2 illustrates the global relationship between labeled adware and malware, where the X-axis presents the adware confidence (AC) and the Y-axis shows the probability of being malicious. When considering $AC = 1$, standing for the most conservative situation, for which families need to contain apps with no other types than adware to fulfill Formula 2. This consideration leads to the worst situation that apps labeled as adware are highly likely to be malware too (with the probability of 99.83%). However, scenarios such as type mislabeling, individual behaviors etc. always happen in practice. Situations of $AC = 1$ could mislead by these noises. So a reasonable error margin should be considered when choosing the value of AC. On the other hand, a turning point has been noticed around $AC = 0.8$. After this point, malware probability increased dramatically. So whether the error margin could be large as 20% is a crucial decision for practitioners to consider.

D. Web Service

Based on the malware probability model, we then implement a web service that 1) regularly updates the model based on the latest app set of AndroZoo; 2) takes a set of Android labeled adware as input and then returns the condemnation results indicating which adware should be considered as malware. We expect this web service to be used as a common means in our community to condemn adware and thus to present consistent, and hopefully more accurate, empirical studies.

IV. Threats to Validity

The threat to the validity of our study mainly lies in the exhaustiveness of our dataset, which may not be representative to the current app ecosystem. However, we have conducted our study on so far the largest app set available in our community in order to mitigate this threat.

Furthermore, the anti-virus labels yielded by VirusTotal may not be perfect. There may be a possibility that a benign app is mistakenly flagged as a malware while a non-adware is flagged as an adware. Besides, since the anti-virus labels are harvested at a certain time, we are not aware of any changes anti-virus may yield. Nevertheless, in this work, we attempt to alleviate this threat by conducting our study on a fairly large number of apps. Therefore, the empirical findings presented in this work will unlikely be impacted.

Finally, the adware and malware families used in this work are categorized via Euphony. Hence, this work shares the same threats to validates of that of Euphony. For example, at the moment, we do not take familial ties (i.e., some adware/malware families may overlap) into consideration, which may result in labeling errors.

V. Related Work

To the best of our knowledge, we are the first one to quantitatively investigate the relationship between adware and malware in the Android research community. However, adware/malware analysis of Android apps has been explored from several aspects [14], [15]. In this section, we highlight some representative ones.

Similar to Euphony, which has been leveraged in this work to infer adware/malware families, AVClass [16] is another tool that can be leveraged to achieve the same purpose. Besides, as mentioned by the authors, AVClass requires a ground-truth list of known families to distinguish from generic tokens, and relies on vendor-specific rules to remove vendor suffixes, which however are not needed by Euphony.

Many state-of-the-art works have focused on identifying ad libraries instead of highlighting Android adware [13], [17]–[20]. For example, Li et al. [13] has revealed 240 ad libraries through a heuristic-based approach. Nevertheless, although the findings of those approaches, being a whitelist of ad libraries, can be leveraged to detect Android adware. Indeed, as empirically reported by Dong et al. [21], [22] recently, some mobile apps even attempt to violate the behavioural policies of ad libraries. Those devious apps attempt to entice app users to click ads (unintentionally in most cases) so as to gain more revenues. If a given Android app has leveraged an ad library from the whitelist, we have reasons to believe that it is an adware. Unfortunately, this approach will introduce a lot of false negatives because on the one hand, the whitelist is not

thorough enough to cover all the available ad libraries while on the other hand whitelist-based approaches also cannot address the challenge of obfuscation where a library could be totally renamed.

The authors of AdRisk [23] have demonstrated that ad libraries may expose security and privacy risks, or even perform malicious behavior such as leaking personal user information. For example, they show that ad libraries may execute untrusted codes that are downloaded from internet sources. Even worse, those untrusted codes could be fetched in an unsafe way, which by itself has caused serious security risks. Demetriou et al. [24] reveals that there are four major channels that are recurrently leveraged by ad libraries to collect private user information. To mitigate this, researchers also explore different ways to prevent such information from being leaked [25], [26] while delivering ad content.

VI. Conclusion

In this work, we have conducted a quantitative study on the relationship between adware and malware. The output of this study is a ranked list of ad families with probabilities of being malicious. Each family has been assigned with a malicious rate, showing the possibility of being a malware, given a random adware from that family. Our study provides a means for security analysts to decide whether a given adware should be considered as malware. Finally, by considering all the investigated apps and the global relationship between adware and malware, we present to the community an off-the-shelf web service that condemns automatically adware with a controlled confidence.

VII. Acknowledgement

This work was supported by the Fonds National de la Recherche (FNR), Luxembourg, under projects CHARACTERIZE C17/IS/11693861 and SPsquared 10621687.

References

[1] Oxford Dictionaries. Adware. https://en.oxforddictionaries.com/definition/adware.

[2] Eric Chien. Techniques of adware and spyware. Dublin, Ireland, Oct 2005. Symantec, VB2005 Conference.

[3] Yajin Zhou and Xuxian Jiang. Dissecting android malware: Characterization and evolution. In *IEEE Symposium on Security & Privacy*, San Francisco, May 2012.

[4] Jameel Qadri, Thomas M Chen, and Jorge Blasco. A review of significance of energy-consumption anomaly in malware detection in mobile devices. 2016.

[5] Yuta Ishii, Takuya Watanabe, Mitsuaki Akiyama, and Tatsuya Mori. Clone or relative?: Understanding the origins of similar android apps. In *Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics*, pages 25–32. ACM, 2016.

[6] Bartlomiej Uscilowski. Mobile adware and malware analysis. Technical report, Symantec, 2013.

[7] Li Li, Alexandre Bartel, Tegawendé F Bissyandé, Jacques Klein, Yves Le Traon, Steven Arzt, Siegfried Rasthofer, Eric Bodden, Damien Octeau, and Patrick Mcdaniel. IccTA: Detecting Inter-Component Privacy Leaks in Android Apps. In *Proceedings of the 37th International Conference on Software Engineering (ICSE 2015)*, 2015.

[8] Andrea Continella, Yanick Fratantonio, Martina Lindorfer, Alessandro Puccetti, Ali Zand, Christopher Kruegel, and Giovanni Vigna. Obfuscation-resilient privacy leak detection for mobile apps through differential analysis. In *Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS)*, pages 1–16, 2017.

[9] Médéric Hurier, Tegawend F. Bissyandé, Yves Le Traon, Jacques Klein, Guillermo Suarez-Tangil, Santanu Kumar Dash, and Lorenzo Cavallaro. Euphony: Harmonious unification of cacophonous anti-virus vendor labels for android malware. In *The 14th International Conference on Mining Software Repositories (MSR)*, Argentina, May 2017.

[10] Kevin Allix, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon. Androzoo: Collecting millions of android apps for the research community. In *Mining Software Repositories (MSR), 2016 IEEE/ACM 13th Working Conference on*, pages 468–471. IEEE, 2016.

[11] Li Li, Jun Gao, Médéric Hurier, Pingfan Kong, Tegawendé F Bissyandé, Alexandre Bartel, Jacques Klein, and Yves Le Traon. Androzoo++: Collecting millions of android apps and their metadata for the research community. *arXiv preprint arXiv:1709.05281*, 2017.

[12] VirusTotal. About page. https://virustotal.com/en/about/. accessed on 1st March 2018.

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

[14] Pingfan Kong, Li Li, Jun Gao, Kui Liu, Tegawendé F Bissyandé, and Jacques Klein. Automated testing of android apps: A systematic literature review. *IEEE Transactions on Reliability*, 2018.

[15] Li Li, Tegawendé F Bissyandé, Mike Papadakis, Siegfried Rasthofer, Alexandre Bartel, Damien Octeau, Jacques Klein, and Yves Le Traon. Static analysis of android apps: A systematic literature review. *Information and Software Technology*, 2017.

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

[17] Annamalai Narayanan, Lihui Chen, and Chee Keong Chan. Addetect: Automated detection of android ad libraries using semantic analysis. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014.

[18] Leonid Glanz, Sven Amann, Michael Eichberg, Michael Reif, Ben Hermann, Johannes Lerch, and Mira Mezini. Codematch: obfuscation won't conceal your repackaged app. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 638–648. ACM, 2017.

[19] Menghao Li, Wei Wang, Pei Wang, Shuai Wang, Dinghao Wu, Jian Liu, Rui Xue, and Wei Huo. Libd: scalable and precise third-party library detection in android markets. In *Software Engineering (ICSE), 2017 IEEE/ACM 39th International Conference on*, pages 335–346. IEEE, 2017.

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

[21] Feng Dong, Haoyu Wang, Li Li, Yao Guo, Guoai Xu, and Shaodong Zhang. How do mobile apps violate the behavioral policy of advertisement libraries? In *Proceedings of the 19th International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2018.

[22] Feng Dong, Haoyu Wang, Li Li, Yao Guo, Tegawendé F Bissyandé, Tianming Liu, Guoai Xu, and Jacques Klein. Frauddroid: Automated ad fraud detection for android apps. In *The 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2018)*, 2018.

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

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

[25] Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. 2010.

[26] Saikat Guha, Bin Cheng, and Paul Francis. Privad: Practical privacy in online advertising. In *USENIX conference on Networked systems design and implementation*, pages 169–182, 2011.