A First Look at Mobile Ad-Blocking Apps

Muhammad Ikram
Data61, CSIRO and UNSW
Sydney, Australia
Email: Muhammad.Ikram@data61.csiro.au

Mohamed Ali Kaafar
Department of Computing, Macquarie University
Sydney, Australia
Email: Dali.Kaafar@mq.edu.au

***Abstract*—Online advertisers, third party trackers and analytics services are constantly tracking user activities as they access web services through their web browsers or mobile apps. While, web browser plugins disabling and blocking Ads (often associated tracking/analytics scripts), e.g. AdBlock Plus[3] have been well studied and are relatively well understood, an emerging new category of apps in the tracking mobile eco-system, referred as the mobile Ad-Blocking apps, received very little to no attention. With the recent significant increase of the number of mobile Ad-Blockers and the exponential growth of mobile Ad-Blocking apps' popularity, this paper aims to fill in the gap and study this new category of players in the mobile ad/tracking eco-system.**

This paper presents the first study of Android Ad-Blocking apps (or Ad-Blockers), analysing 97 Ad-Blocking mobile apps extracted from a corpus of more than 1.5 million Android apps on Google Play. While the main (declared) purpose of the apps is to block advertisements and mobile tracking services, our data analysis revealed the paradoxical presence of third-party tracking libraries and permissions to access sensitive resources on users' mobile devices, as well as the existence of embedded malware code within some mobile Ad-Blockers. We also analysed user reviews and found that even though a fraction of users raised concerns about the privacy and the actual performance of the mobile Ad-Blocking apps, most of the apps still attract a relatively high rating.

I. Introduction

Online advertising is ubiquitous in today's digital economy. Embedded third-party tracking libraries in websites and mobile applications (apps in short) is common and perform a variety of functions ranging from enhancing user experience, social sharing to the monetisation of services by enabling targeted and location-based advertisements. Such integration has evolved into a tangled eco-system illustrated by the top10K Alexa websites each integrating on average more than 30 different third-party tracking services for user profiling and advertisement purposes [25]. Likewise, the popularity of mobile apps resulted into a thriving mobile tracking and advertising ecosystem that is perhaps more privacy-invasive, due to the ubiquitous nature of the mobile apps usage.

Sensitive user data, including contacts, locations, SMS and web history is readily accessible on mobile devices for apps geared with the relevant permissions, and represent a valuable asset for third-party advertisers and trackers, which in turn poses serious privacy and security risks beyond the discomfort that mobile ads displayed on rather small screens of smartphones might generate. Naturally, the mobile apps ecosystem has recently witnessed the emergence of a new class of Ad-blocking[1] tools, packaged as mobile apps, in popular mobile app stores such as Google Play.

978-1-5386-1465-5/17/$31.00

This paper presents the first characterisation study of Android mobile Ad-Blocking apps with a focus on security and privacy offered by these apps. In particular, we analyse the Android permissions mobile Ad-Blockers request and we perform static analysis of the code to investigate the presence of malware and third party tracking libraries.

We collect and extract from a corpus of more than 1.5 million Android apps, 97 mobile apps for which the name or the description suggest they enable to either block ads or to block trackers. We then manually check that the apps actually fall into the category of Ad-Blocking apps (cf. Section II).

We use a set of tools to decompile the Ad-Blocking apps and analyse the source code of each of the mobile Ad-Blockers. We then inspect the apps to reveal the presence of third-party tracking libraries and sensitive permissions for critical resources on users' mobile devices. According to VirusTotal[2]'s classification, we observe instances of Ad-Blockers using excessive advertising, displaying full-screen advertisements windows and embedding malware in the source code (cf. Section III).

This paper makes the following major contributions:

- **Mobile Ad-Blockers identification and taxonomy:** We investigate a dataset of 1,554,253 Android apps on Google Play and identify 97 Ad-Blocking apps. Based on the intended functionality and an inspection of the code, we provide a generic taxonomy of Ad-Blockers depending on the mechanisms used to block ads and the (corresponding) tracking/analytics services.
- **Third-party user tracking:** We perform static analysis on mobile Ad-Blockers' source code. We observe that, albeit Ad-Blocking apps' claims to block ads and prevent tracking, 68% of them still embed third-party tracking and ads libraries in their code, potentially leaking personal information to third-parties. We also observe that 24% display ads.
- **Sensitive Permissions Access:** Our analysis reveal that 89% of the Ad-Blockers request sensitive permissions to access critical resources such as user contacts, accounts, text messages, and user browsing history.

[1]We use the term Ad-Blocking to refer to apps blocking both ads and tracking/analytics services.

[2]An Anti-virus (AV) tools conglomerate, https://virustotal.com

- **Malware presence and inefficiency of Ad-Blockers:** According to VirusTotal scan reports, 13% of the Ad-Blockers have malware presence in their source code with instances of spyware and adware to spy on users' behaviour. We analyse user reviews on Google Play to sense whether users expressed concerns about the security, the privacy, and the inefficiency–in term of not blocking ads and trackers–of the Ad-Blockers. Our analysis reveals that albeit users have publicly raised concerns in their app reviews yet some of the Ad-Blockers have high ratings and a significant number of installs.

II. Characterizing Ad-Blockers on Google Play

We first briefly introduce mechanisms used by Mobile Ad-Blocking apps to block trackers and ads. Then, we describe our method for identifying Ad-Blockers on Google Play and show our characterisation of the collected Ad-Blockers.

1) Android Ad-Blocking Mechanisms: Third-party advertisers and trackers use HTTP(s) to deliver Ads to end users. Ad-Blockers intercept and filter ads- and tracking-related traffic. To this end, Ad-Blockers can employ filters either as browsers' extensions (add-ons) or as a built-in functionality (*Case 1* in Figure 1) to block ads and trackers from the traffic generated while the user (or associated app) accesses the Internet. Ad-blockers can also implement mechanisms such as virtual private networks (VPN) tunnels[3] to intercept and filter ads-related traffic, either locally on mobile devices (Case *2* in Figure 1) or on remote servers (Case *3* in Figure 1), from all installed apps.

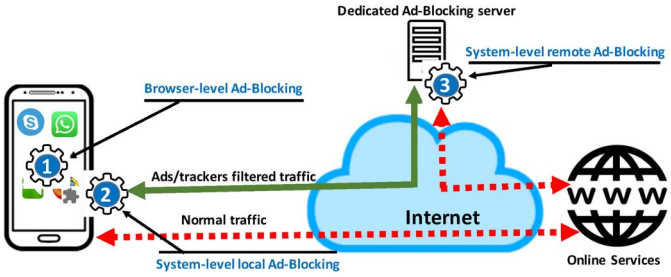


Fig. 1: Overview of Android's Ad-Blocking mechanisms: (1) *brower-level Ad-Blocking*–browser with built-in functions or add-ons to block third-party advertisements and trackers on browsed webpages, (2) *system-level local Ad-Blocking*–specialized Android apps which intercept all installed apps' traffic and filter-out advertisements and trackers, (3) *system-level remote Ad-Blocking*–specialized Android apps which forward user traffic to dedicated servers for Ad-Blocking.

2) Dataset in Use: In order to identify as many Ad-Blockers as possible on Google Play, we implemented a Google Play crawler that takes advantage of two complementary seeds: *(i)* the top 100 apps from four Google Play categories likely to contain Ad-Blockers: Tools, Communication, Personality, and Productivity; and *(ii)* the list of apps by querying Google Play's search with Ad-Blocking-related keywords. The keywords used are: "Ad-Block" (and variants including "ads block", "ad block", "ads-blocking", "blocking-ads" etc.), and "privacy", "tracking-free", "Anti-tracking" in their app descriptions. With these seeds, our crawler follows a breadth-first-search approach for any other app considered as *"similar"* by Google Play and for other apps published by the same developer. In total, we surveyed 1,554,253 apps for a 4-week period in December 2016. Finally, by searching Ad-Blocking keywords in the descriptions of 1,554,253 apps, followed by manual inspection[4], we identified 97 Ad-Blockers (87 free and 10 paid apps).

Android apps are typically written in Java code (and possibly with some additional native code). The overall Java code–implementing the intended functionality and third-party library for extended service or feature–is compiled to a `.dex` file, containing compressed bytecode that runs in the Dalvik virtual machine. Android apps are distributed on marketplaces such as Google Play store as `.apk` files, which bundle the `.dex` code with the app's specification file named `AndroidManifest.xml`. To download the apps' `.apk` files (*aka* APKs) and other apps' metadata from Google Play (*e.g.,* app description, number of installs, developer information, user reviews and app rating), we use Google Play Unofficial Python API [14] for free apps and the Raccoon APK Downloader for paid apps [18]. Finally, we use `ApkTool` and `dex2jar` to decompile and extract each app's source code and the corresponding `AndroidManifest.xml`.

For a baseline comparison (cf. Section III), we also collected 500 randomly selected free non-Ad-Blocking apps–collected from Tools, Communication, Personality, and Productivity categories–from Google Play. In order to further analyse ad-blockers and reproduce our findings, the dataset and crawling scripts are available upon request.

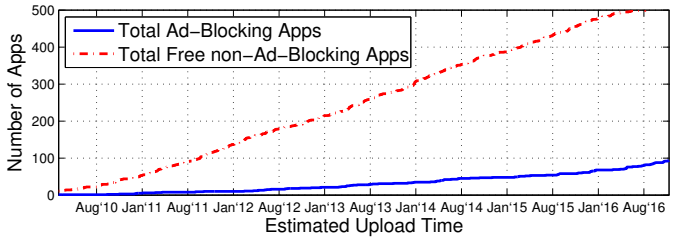
3) The Rise of Android Ad-Blockers: We start our analysis by observing the increase of Ad-Blockers available for download on Google Play over time. Given that Google Play does not report the actual release date of the apps but their last update, we use the date of their first review as a proxy of their release date. For those Ad-Blockers without user reviews, we approximate the release date with the date of their last update.

Figure 2a shows the steady increase of Ad-Blocking apps listed on Google Play. During the 3-year period that spans between August 2012 and August 2015, the number of Ad-Blocking apps increased 3-fold. Overall, the analysed Ad-Blockers receive high user ratings: 41% of the Ad-Blockers have more than 100K installs and 68% of them have at least a 4-star rating as shown in Figure 2b. We cannot tell whether the installs and the reviews are legitimate or if those ratings were actually acquired by the app developers, using apps' promotion

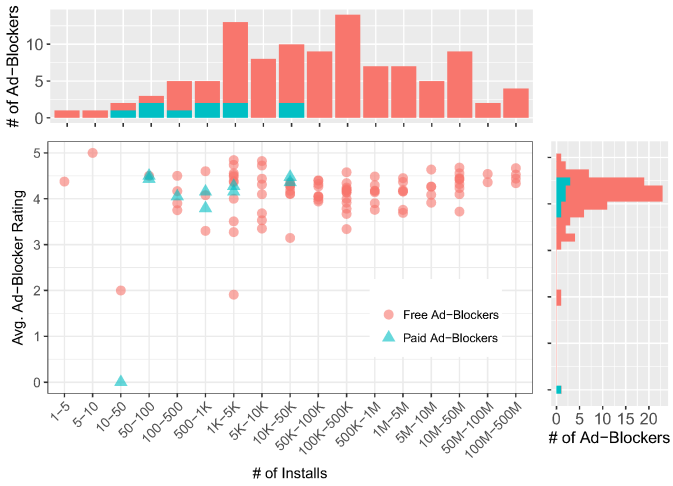
[3]https://developer.android.com/reference/android/net/VpnService.html

[4]We manually checked their description to ensure they do belong to what users would consider as an Ad-Blocking tool or alternatively a tool to block tracking.

services on underground marketplaces to promote the apps[5].



(a)



(b)

Fig. 2: Evolution of Ad-Blockers on Google Play: (2a) Number of Ad-Blockers available on Google Play over time. (2b) Distribution of average rating vs. installs per Ad-Blocker.

Ad-Blockers' Classification.

Given that Ad-Blockers block ads and trackers in a different fashion, we aim at categorising Ad-Blockers according to their intended functionality (or complementary features) and their Ad-Blocking mechanism. Unfortunately, Google Play's app categories (e.g., Tools and Communication) are too broad to capture the actual purpose and functionality of the app. Moreover, apps may not include any detail on their Ad-Blocking mechanisms in their descriptions on Google Play. To identify Ad-Blocking mechanisms of the analysed Ad-Blockers, we manually tested and categorised them into two classes, listed in Table I.

App Category	% of Apps (N = 97)
Browser-level Ad-Blockers	86%
System-level or VPN-based Ad-Blockers	14%

TABLE I: Classification of Ad-Blockers.

We found that 86% of the analysed Ad-Blockers have built-in Ad-Blocking mechanism to block ads and trackers on a given webpage. These Ad-Blockers do not block In-App ads or Ad-banners (cf. Case (*ii*) in Figure 3). On the other hand, 14% of the apps create local or remote VPN-tunnel for blocking

[5]https://www.seoclerk.com/Link-Building/192193/Test-Your-iOS-or-Android-Apps-On-Smartphone-And-Provide-Review-And-Rating

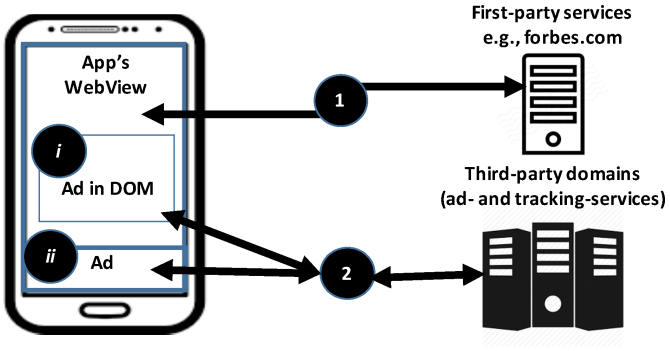


Fig. 3: Overview of ads displayed in Android apps. Here, an app accesses, *step 1*, first-party webpage. Depending upon third-party libraries embedded in app's source code and third-party JavaScript programs included in the webpage's HTML source code, the app sends additional requests, *step 2*, to third-party ad services. The app displays ads from third-party ad services either in In-App Ads-banner, *Case ii*, in a browsed webpage, *Case i*, or both.

ads-related traffic, whether ads appeared in ad-banners or in a browsed webpage (cf. Case *(i) & (ii)* in Figure 3), from all installed apps.

Geographical Distribution of Ad-Blockers.

Here, we are interested in characterising the distribution and the popularity of Ad-Blockers per country and per geographical regions. To this end, we rely on the geographical popularity or *rank* data from AppAnnie [7]. AppAnnie offers downloads or installs analytics to apps' developers and gathers apps' longitudinal usage from app markets such Google Play and iTunes. It uses developers' accounts to tracks apps' downloads from app markets and assigns numeric values or *ranks* to represent apps' popularity in each geographical region or country. For each Ad-Blocker, we use AppAnnie API to obtain rank values per country. From the collected rank data, we map Ad-Blockers to countries and count the number of distinct Ad-Blockers per country. Moreover, we measure the median ranks to determine the popularity of Ad-Blockers per country.

Figure 4 shows the cumulative distribution of the number of countries with the number of distinct Ad-Blockers according to AppAnnie countries rank dataset. We observe a significant difference in the geographical coverage among the classes of Ad-Blockers. The distribution suggests that paid Ad-Blockers have more geographically scattered downloads around the globe: 80% of paid Ad-Blockers have downloads in more than 20 countries where as the 56% of the free counterparts have their users located in less than 20 different countries. The figure also reveals that browser-level Ad-Blockers have a higher geographical coverage in terms of installs when compared to systems-level or VPN-based Ad-Blockers.

We also observe a significant difference in the number of Ad-Blockers per geographical region. Several countries including China, Macedonia, and Lativa have diverse set of free Ad-Blockers–each has 56% of distinct Ad-Blockers.

Maxthon Browser, a browser-based free Ad-Blocker, has more than 10M installs in 93 different countries, suggesting its global popularity. On the other hand Piggy Browser, also a free Ad-Blocker, has 1K installs only in a single country, Japan.

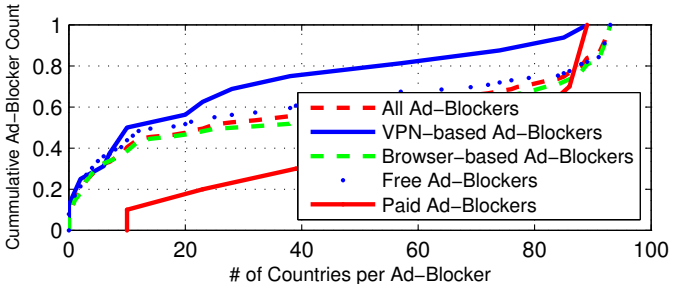
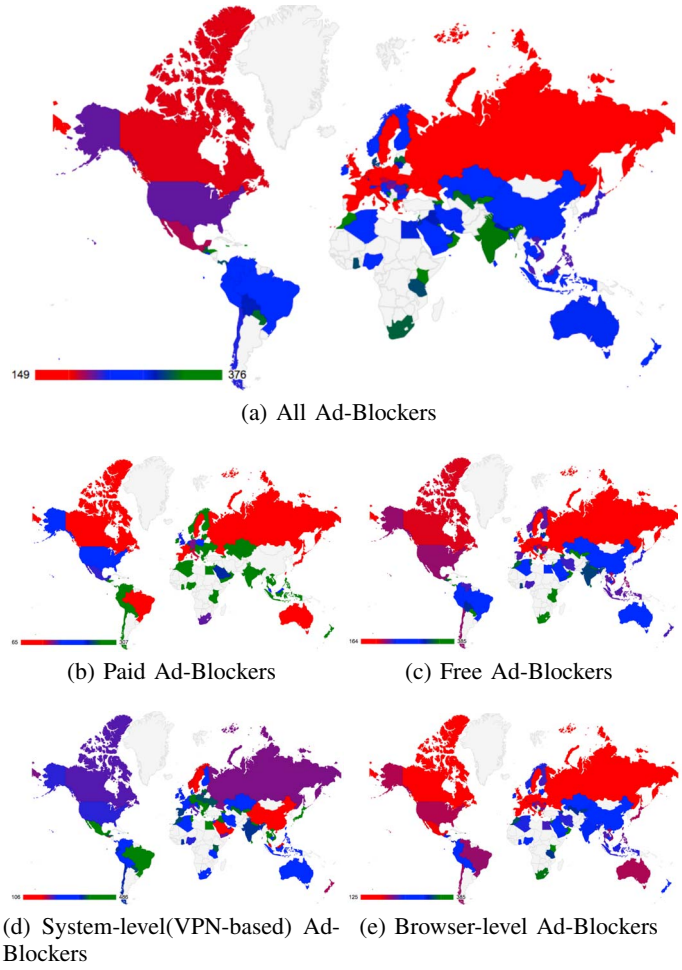


Fig. 4: Geographical usage (or installs) of Ad-Blockers.

The number of Ad-Blockers per country does not reveal the actual popularity of Ad-Blockers per country. We further analyse the popularity of each Ad-Blocker per country by measuring the median rank of all Ad-Blockers per country. Figure 5 shows the median rank of the analysed Ad-Blockers. Notably, VPN-based Ad-Blockers are more popular in countries such as China (median rank = 106), Macau (141), Saudi Arabia (152) and Oman (161) whereas browser-based Ad-Blockers are popular in France (125), Ukraine (132), and Germany(145). Compared to free Ad-Blockers, paid Ad-Blockers are popular in Taiwan (65), Spain (70), and Japan (70). When considering the median ranks of all Ad-Blockers per country, we observe that Ad-Blockers are popular in France (149), Russia (170), and Germany (176).



(a) All Ad-Blockers

(b) Paid Ad-Blockers

(c) Free Ad-Blockers

(d) System-level(VPN-based) Ad-Blockers

(e) Browser-level Ad-Blockers

Fig. 5: An overview of Ad-Blockers' popularity per country. The lower values of median rank, dark red colour, represent that Ad-Blockers are more popular in the corresponding geographical region or country.

III. Analysis of Ad-Blockers

In this section, we analyse the source code of the Ad-Blockers using static analysis. We examine the filter lists used by the Ad-Blockers and investigate the Ad-Blockers requesting sensitive permissions. Further, we evaluate the Ad-Blockers for the presence of third-party ads and tracking libraries, embedded in their code, and study their malicious activities using VirusTotal. Finally, we present our analysis of Ad-Blockers' reviews on Google Play and report on users concerns about potential security and privacy issues as well as the inefficiency of Ad-Blockers.

# Filter Lists	Ad-Blockers		
	Paid	Free	All
1	70%	84%	82%
2	30%	8%	10%
≥3	0%	8%	8%

TABLE II: Distribution of filter lists used by the analysed Ad-Blockers.

A. Filter Lists in Use

Ad-Blockers usually employ filter lists (i.e. Black lists) either downloaded locally or present on a remote server to block (or allow) third-party ads and tracking services. Using `Apktool`, the decompilation of Ad-Blockers' APKs allows us to reveal the employed filter lists. First, we decompile the Ad-Blockers' APKs and then search for specific keywords in the source code. The keywords include: "easylists", "abp", "hosts", "mvps", "blacklist", "blocklist", "whitelist", "exception rules", and "acceptable ads". Finally, for each Ad-Blocker, we use `JD-GUI` to manually inspect the decompiled source codes and to determine the employed filter lists.

In Table II, we observe that 82% of the analysed Ad-Blockers contain only one filter list. While 18% of the Ad-Blockers use a composition of several filter lists.

Table III shows the distribution of filter lists used by the Ad-Blockers. 90% of the paid Ad-Blockers and 39% of the free Ad-Blockers employ custom-built filter lists to block ads and tracking related traffic. We found that 16% of the Ad-Blockers use EasyList [9] to filter advertisements on a browsed webpage. Our manual inspection of the code using `ApkTool` and `JD-GUI` reveals that all VPN-based Ad-Blockers (cf. Section II-1) such as F-Secure Freedome VPN and DashVPN use their custom-built filters lists to block ads and trackers. In particular, F-Secure Freedome VPN app blocks any traffic

Filter List	Ad-Blockers			Description
	Paid	Free	All	
CustomizedList	90%	39%	42%	Customized list for ads/trackers
EasyList [9]	0%	18%	16%	Filter list of trackers/ads
hpHosts [15]	10%	10%	10%	Filter list of ad/tracker/malicious hosts
AcceptableAdsRules [5]	0%	7%	7%	List of non-intrusive, acceptable ads
MalwareHosts [16]	0%	5%	5%	Filter list of malicious hosts
FanboySocial [13]	0%	4%	4%	Filter list of social buttons/widgets
AntiAdblockRules [1]	0%	4%	4%	Filters for evading anti-Adblock scripts
HostStevenBlack [20]	0%	3%	3%	Filter list of ad/tracker/malicious hosts
EasyPrivacy [11]	0%	3%	3%	Filter list of trackers/analytics
AdSweep [4]	0%	3%	3%	List of Rules to hide ads on websites
EasyListChina [8]	0%	2%	2%	EasyList tailored for Chinese websites
YoyoHosts [21]	0%	2%	2%	Filter list of ad/tracker/malicious hosts
SomeOneCareHosts [19]	0%	1%	1%	Filter list of ad/tracker/malicious hosts
MvpsHosts [17]	0%	1%	1%	Blacklist of ad/tracker/malicious hosts
EasyListNEHide [10]	0%	1%	1%	EasyList without element hiding rules
AdAwayHosts [2]	0%	1%	1%	Filter list of mobile ad/tracker hosts

TABLE III: Distribution of all filter lists in the analysed Ad-Blockers.

associated with web and mobile tracking including Google Ads, DoubleClick, and other popular tagging/analytics services such as Google Tag and ComScore [12]. Moreover, 3% of the Ad-Blockers such as Simple FLV and Fast Browser inject JavaScript codes, AdSweep [4], to hide rather than block ads on a browsed webpage.

B. Permission Analysis

We investigate how Ad-Blockers request Android permissions to access sensitive system resources. For each Ad-Blocker, we extract the requested permissions by parsing *uses-permission* and *service* tags in the `AndroidManifest.xml`. We exclude network-related permissions like Internet access which are inherent to Ad-Blockers.

Figure 6 compares the permissions requested by Ad-Blockers with the baseline (cf. Dataset in Section II), which we included for reference. We use the method-to-permission mapping provided by Au *et al.* [22] to investigate the source-code segments invoking the methods protected by each Android permission. Our analysis reveals that Ad-Blocking apps request access to permissions rarely requested by free non-Ad-Blocking apps.

Apps such as Anti-virus apps request the `READ_LOGS` permission to inspect other apps' activities [6][27]. However, we observe that Ad-Blockers like UC Browser and DU Browser also request access to it. Android's documentation [6] flags the `READ_LOGS` permission as highly sensitive as app developers may carelessly misuse Android's logging capabilities and (unintentionally) expose personal information (including passwords) to any other apps requesting it.

Several other permissions listed in Figure 6 appear unusual requirements for Ad-Blockers. For each case, we manually checked the legitimacy of these requests without finding a clear evidence of a deliberate abuse of the granted permissions. However, we found that spyware Ad-Blocking apps (which we further investigate in Section III-D) request the `READ_SMS` permission to read text messages whereas AV apps may use it to scan text messages for possible malware presence. Similarly, apps requesting `READ_CONTACTS` incorporate functions in the likes of blocking text and calls from specific phone numbers or sharing features through SMS or email.

# Trackers	Ad-Blockers			Free non-Ad-Blocking Apps
	Paid	Free	All	
0	40%	31%	32%	16%
1	50%	14%	18%	7%
2	10%	13%	12%	13%
3	0%	10%	9%	16%
4	0%	11%	10%	14%
$\geq$5	0%	21%	19%	34%

TABLE IV: Analysis of third party ads and tracking libraries in Ad-Blockers and free non-Ad-Blocking apps.

C. Embedded Ads and Tracking Libraries

We examine the presence of embedded libraries for tracking or advertising purposes in the source-code of each Ad-Blocker. In order to conduct our analysis, we use `ApkTool` to decompile each Ad-Blocker and perform a dictionary-based search for ads and tracking libraries inside the decompiled source code. Build upon previous work on mobile third-party libraries' characterization [30], we compile a comprehensive dictionary of 338 mobile third-party tracking libraries.

Table IV compares the number of ads and tracking libraries used by Ad-Blocking apps with the presence of ads and tracking libraries in the reference set of 500 free non-Ad-Blocking apps. We observe that 68% of the Ad-Blocking apps embed at least one third-party ads and tracking library in their code. The penetration of tracking libraries in Ad-Blocking apps is however significantly lower than in the reference set of 500 non-Ad-Blocking apps with 84% of the latter having at least one embedded tracking library.

Since Ad-Blockers intend to block trackers and (intrusive) advertisements, the lower presence of tracking and advertisement libraries is actually meaningful. Nevertheless, we identified at least one targeted ads library in 68% of the Ad-Blockers claiming (as mentioned in their apps description on Google Play) to block advertisements. While paid apps are often thought to be free from ads (as their business model is supposedly not driven by advertising), we observed a disturbing 60% of the paid Ad-Blockers having at least one embedded third-party ad and tracking library. In particular, Photon Flash Player & Browser and Perk Browser–two popular apps, which combined have more than 11M installs–have the highest number of embedded third-party tracking libraries: 13 and 11 tracking libraries respectively. In general, 19% of the Ad-Blockers have at least five third-party ads and tracking libraries.

Figure 7 ranks the trackers in all analysed Ad-Blocking apps. Google Ads and Facebook social analytics are the most popular ones among our corpus of Ad-Blockers. A closer examination at the long-tail of the distributions in Figure 7a and Figure 7b reveal that the least popular third-party ands tracking libraries, respectively, are more common in Ad-Blockers than in free non-Ad-Blocking apps. For instance, Ad-Blockers like Opera Browser and DU Browser, each has over

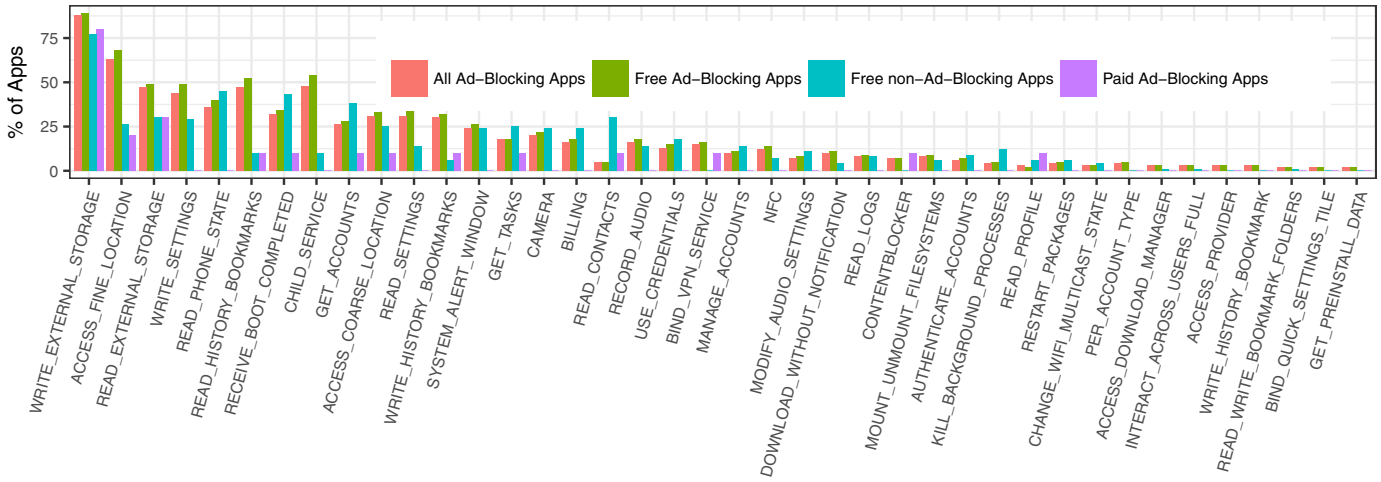
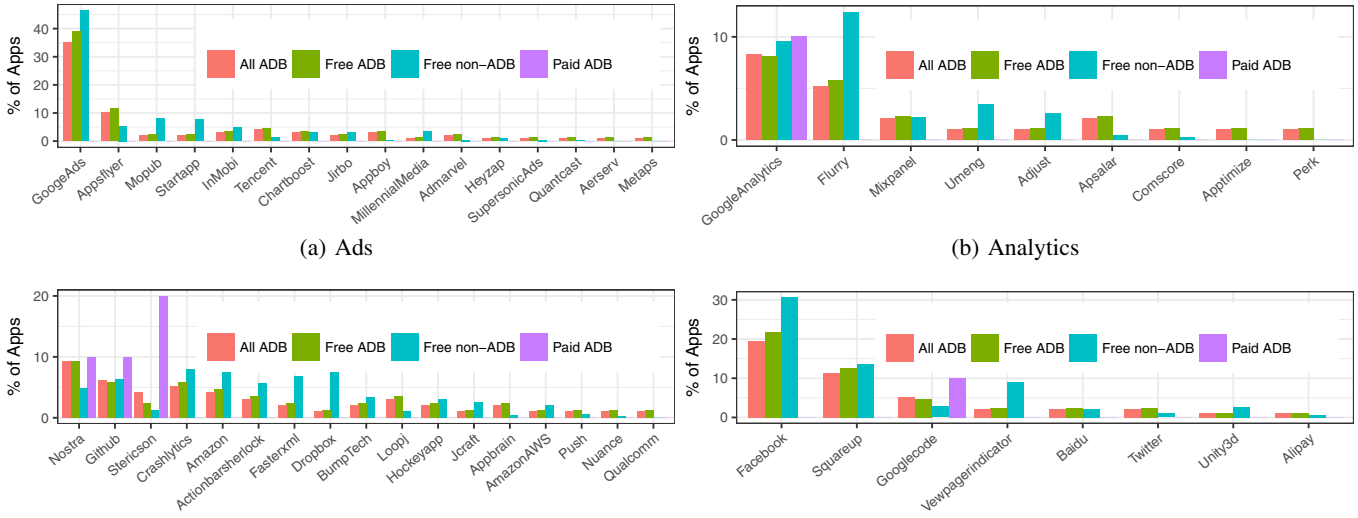


Fig. 6: Detailed comparison of Android permissions (x-axis) requested by Ad-Blocking and free non-Ad-Blocking apps.



(a) Ads

(b) Analytics

(c) Utility

(d) Misc. (Payment, Game Engine, Development Aid, UI component, and Social Network)

Fig. 7: Third-party libraries (x-axis) in Ad-Blocking (ADB) and free non-Ad-Blocking (non-ADB) apps.

10M installs, integrate libraries like Appboy[6] and Tencent[7] for tracking and delivering targeted ads.

D. Malware Analysis

We explore the presence of malware in the analysed Ad-Blockers. In order to effectively identify any malware activity on mobile apps, it is critical to rely on multiple malware scanning tools as malicious components may be designed to circumvent some AV tools and malware scanners. To improve the confidence of our malware scans, we rely on VirusTotal, an online service that aggregates the scanning capabilities provided by multiple AV tools.

We automate our malware scanning process by using VirusTotal's public API. After completing the scan, VirusTotal generates a report that indicates which of the participating AV scanning tools detected any malware activity in the app and the corresponding malware signature (if any). Given that a scanning tool may produce false positives, we rely on the "AV-rank" metric (*i.e.,* the number of affiliated AV tools that identified any malware activity) to reason about the maliciousness of an Ad-Blocker. We consider an "AV-rank" threshold ≥ 3 as a signal for malware presence in Ad-Blockers. Additionally, we glean developers' metadata from Google Play and AppAnnie to investigate the geolocations of developers of malicious Ad-Blockers.

#	App ID	Price	Rating	Installs	AV-rank	DevLoc
1	Deep Search Browser	Free	2.6	1K	10	PK
2	sFly Network Booster	Free	4.3	1K	10	CN
3	Faster Browser Ever	Free	4.0	1K	6	RU
4	FastCat	Free	4.5	100	6	AE
5	Magneto Browser	Free	4.0	100	4	IN
6	Adskip Browser	Free	2.0	5K	4	CN
7	Maxthon Browser	Free	4.4	10M	3	CN

TABLE V: Ad-Blockers with a VirusTotal AV-rank ≥ 3. DevLoc represents the geolocation of Ad-Blockers' developers.

Table V lists the Ad-Blocking apps ranked by their AV-rank. We include developers' geolocation and apps' Google Play

[6]https://www.appboy.com

[7]http://tencent.com/en-us/ps/adservice.shtml

rating and the number of installs for each app for reference. 21% of the analysed Ad-Blocking apps have at least one positive malware report according to VirusTotal with 7% of the Ad-Blockers have an "AV-rank" above our threshold. The malware signatures correspond to five different classes of malware: Adware (22%), Trojan (23%), Malvertising (16%), and Riskware (39%). We observe that malware developers are located in Russia and Asian countries such as India, China, Pakistan, and United Arab Emirates.

Next, we analyse the permission-protected API access of the malicious Ad-Blockers. Maxthon Browser incorporates Adware on its source-code and requests the intrusive `SYSTEM_ALERT_WINDOW` permission which allows the Maxthon Browser to draw window alerts (e.g., full screen Ad windows) on top of any other active app. Likewise, DU Browser, which incorporates Riskware elements according to VirusTotal, requires the `READ_LOGS`, `READ_PHONE_STATE`, `READ_HISTORY_BOOKMARKS`, `READ_SETTINGS`, and `WRITE_SETTINGS` permissions to read users' settings, hijack bookmarks, change browser's start page, and link web searches to potentially lower ranked sites.

E. Apps Reviews Analysis

We use users' negative comments to capture the perceptions and concerns about the Ad-Blocking functionality of the analysed Ad-Blockers. Our reasoning to focus our analysis on negative reviews, 1- and 2-star reviews appeared on Google Play, for popular apps is that any serious Ad-Blocking-related concern exposed by a user should receive a negative review.

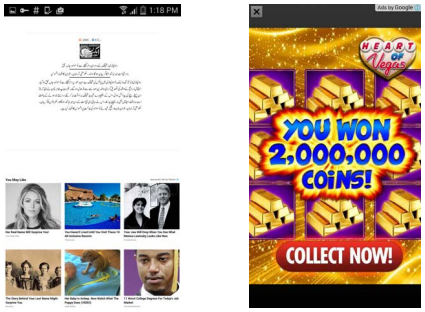
Complaint Category	**% of negative reviews** ($N = 20,008$)
1 Star Reviews	13,764
2 Star Reviews	6,244
Allowing/not-blocking Ads	16%
Bugs & battery-life	7%
Abusive permissions	1.5%
Malware/fraud reports	0.65%

TABLE VI: Classification of negative reviews for Ad-Blockers in Google Play.

To better identify whether users publicly report any Ad-Blocking concerns after using each Ad-Blocker, we analyse (with manual supervision) the content of 20,008 negative reviews for 88% of the analysed Ad-Blockers[8]. We label the app reviews into the 4 categories listed in Table VI which cover from performance concerns and bugs to different types of Ad-Blocking concerns as well as abusive or intrusive permission requests. We exclude from our analysis any reviews related with usability concerns such as bugs and crashes. Note that 7% of the complaints are about crashes and other performance aspects such as bugs and battery-life overhead.

We observe that 16% (=3,201) of the negative reviews report on the allowing or displaying ads in In-App Ad-banner or visited websites (cf. Figure 3). We found that 77% of the analysed Ad-Blockers have at least one negative review

[8]12% of the Ad-Blockers do not have negative reviews or do not have reviews at all.



(a) (b)

Fig. 8: Screenshots showing inefficiency of DashVPN, a system-level Ad-Blocker: (8a) DashVPN does not block clickbaits from Taboola (`taboola.com`) and (8b) shows full-screen ads from Google.

about their inefficiency to block ads. We install and manually check the negatively reviewed apps and found that 24% of the apps show ads and display full-screen advertisements (e.g., Figure 8).

Notably, 1.5% of the negative reviews report on the intrusiveness and sensitive permissions requests of 37% of the analysed Ad-Blockers. We found that these 37% Ad-Blockers have at least one third-party ads and tracking library embedded in their code (as per the analysis in Section III-C). Moreover, we observed that 130 ($\simeq$ 0.65%) of the reviews explicitly report malware or fraudulent activity in 22% of the analysed Ad-Blockers, listed in Table VII. Six of the apps reported as malicious by end users are also considered as malicious by VirusTotal (cf. Section III-D). Interestingly however, and despite the presence of some awareness and public comments about the potential security and privacy issues of Ad-Blockers, these apps still exhibit relatively high average ratings and a significant number of installs (cf. Table VII). Our analysis suggests that a significant number of Ad-Blockers do not fulfill their main "intended" Ad-Blocking functionality.

#	App	Class	NR-Ratio	Rating	Installs	AV-positive
1	UC Browser - Fast Download	Free	32%	4.5	100M	✓
2	Opera browser - latest news	Free	15%	4.3	100M	
3	Yandex Browser for Android	Free	12%	4.5	10M	✓
4	Maxthon Browser - Fast&Secure	Free	21%	4.4	10M	✓
5	Photon Flash Player & Browser	Free	28%	3.7	10M	
6	CM Browser - Fast & Light	Free	15%	4.5	50M	
7	Dolphin - Best Web Browser	Free	25%	4.5	50M	
8	Adblock Browser for Android	Free	26%	4.1	5M	✓
9	APUS Browser - Fast Download	Free	22%	4.5	5M	✓
10	Free Adblocker Browser	Free	8%	4.4	1M	
11	Dolphin Jetpack - Fast & Flash	Free	31%	4.1	1M	
12	F-Secure Freedome VPN	Free	11%	4.3	1M	
13	Dolphin Zero Incognito Browser	Free	17%	4.2	1M	
14	Adblock Plus (Samsung Browser)	Free	37%	3.9	500K	
15	Mercury Browser for Android	Free	19%	4.3	500K	
16	Ghostery Privacy Browser	Free	21%	4.1	500K	
17	Rocket Browser	Free	18%	4.4	100K	
18	AC Browser	Free	6%	4.5	100K	
19	NetGuard - no-root firewall	Free	19%	4.2	100K	
20	Dee Browser	Free	9%	4.1	50K	✓

TABLE VII: List of Ad-Blockers that are considered as malicious by users in Google Play reviews and by VirusTotal (AV-positive column). For each Ad-Blocker, the *NR-Ratio* represents the ratio of the number of negative users' comments to the total number of all users' comments.

IV. Related work

The Web has evolved, with increase in the prevalence and complexity of tracking mechanisms since 1996, into a tangled mass of third-party tracking domains embedded into first-party webpages [29]. Research studies have reveal that the top 5% of webpages embed 100 third party domains [32]. Among other services such as provision of multimedia services via content delivery networks and user-interactions, these third parties provide a variety of services such as tracking users, serving ads, and performing site analytics.

Several studies highlighted the privacy risks associated with Android apps over-requesting Android permissions for third-party tracking, advertising and analytic services [30] using techniques like static analysis [22], taint analysis [24], and OS modifications [28]. Previous research also adapted techniques for malware detection such as signature analysis [23] to the mobile context in order to identify potential malicious activity of mobile apps. Using static code and dynamic analysis techniques, Ikram et al., [27] measured mobile VPN apps and identified several security and privacy issues in 283 different VPN permission-enabled Android apps. The authors also highlighted an alarming mismatch in apps' descriptions on Google Play and their actual functionalities.

Studies have examined the effectiveness of web Ad-Blockers [31] [26]. Ikram et al., [26] evaluated the effectiveness of five different web Ad-Blocking plugins and proposed an improved machine-learning based solution to strike the balance between blocking tracking/advertising domains and allowing domains that serve useful content such as CDNs. Wills and Uzunoglu [32], investigated the default and fully configured settings of Ad-Blocking plugins. They observed that the default as well as the fully configured settings, with composite filter lists, of the plugins are inefficient to block ads- and tracking-related traffic. In contrast to the previous work, this paper present the first characterisation study of mobile Ad-Blocking apps with a focus on security and privacy offered by these apps.

V. Conclusion and Future Work

The increasing number of mobile Ad-Blockers available on apps' markets such as Google Play and the growing number of complaints raised by users indicate serious ineffectiveness or usability issues thus necessitate the urge to analyse this unexplored eco-system. The average mobile user rates Ad-Blockers positively even when they have malware presence. According to our study, 16% of negative reviews are related to (or concerned with) the ineffectiveness of the Ad-Blockers suggesting serious performance issues. Moreover, our analysis of Ad-Blockers, reviewed by users, reveals that several Ad-Blockers such as F-Secure Freedome VPN caused several usability issues while running other installed apps or surfing the web. We believe that our work could be extended to study the (in)effectiveness of the Ad-Blockers. As a future work, we plan to complement the insights provided by our analysis with a comprehensive set of active tests to reveal the runtime behavioural aspects of the Ad-Blockers.

References

[1] AakList (Anti-Adblock Killer). https://github.com/reek/anti-adblock-killer/blob/master/anti-adblock-killer-filters.txt.
[2] AdAway Hosts. https://adaway.org/hosts.txt.
[3] Adblock Plus Plugin. https://adblockplus.org.
[4] AdSweep. https://web.archive.org/web/20121126071410/http://www.adsweep.org/.
[5] Allow non-intrusive advertising. https://easylist-downloads.adblockplus.org/exceptionrules.txt.
[6] Android Permissions. http://developer.android.com/guide/topics/security/permissions.html.
[7] App Annie Insights. https://www.appannie.com/en/insights/.
[8] China+EasyList. https://easylist-downloads.adblockplus.org/easylistchina+easylist.txt.
[9] EasyList. https://easylist.to/easylist/easylist.txt.
[10] EasyList Without Element Hiding Rules. https://easylist-downloads.adblockplus.org/easylist_noelemhide.txt.
[11] EasyPrivacy. https://easylist.to/easylist/easyprivacy.txt.
[12] F-Secure Freedome Anti-Tracking Feature Explained. https://community.f-secure.com/t5/F-Secure/F-Secure-Freedome-Anti-Tracking/ta-p/52153.
[13] Fanboy's Social Blocking List. https://easylist-downloads.adblockplus.org/fanboy-social.txt.
[14] Google Play Unofficial Python API. https://github.com/egirault/googleplay-api.
[15] hpHosts. http://www.hosts-file.net.
[16] Malware Domain Hosts Lists. https://www.malwaredomainlist.com/hostslist/hosts.txt.
[17] MVPS Hosts Lists. http://winhelp2002.mvps.org/hosts.txt.
[18] Raccon APK Downloader. http://www.onyxbits.de/raccoon.
[19] SomeOneCare Hosts Lists. http://someonewhocares.org/hosts/hosts.
[20] Steven Black Hosts. https://github.com/StevenBlack/hosts.
[21] YoYo Ad Server List. https://pgl.yoyo.org/adservers/.
[22] K. W. Y. Au, Y. F. Zhou, Z. Huang, and D. Lie. PScout: Analyzing the Android Permission Specification. In *CCS*, 2012.
[23] A. Bose, X. Hu, K. G. Shin, and T. Park. Behavioral Detection of Malware on Mobile Handsets. In *ACM MobiSys*, 2008.
[24] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. TaintDroid: An Information Flow Tracking System for Real-Time Privacy Monitoring on Smartphones. *CACM*, 2014.
[25] S. Englehardt and A. Narayanan. Online Tracking: A 1-million-site Measurement and Analysis . CCS, 2016.
[26] M. Ikram, H. J. Asghar, M. A. Kaafar, B. Krishnamurthy, and A. Mahanti. Towards seamless tracking-free web: Improved detection of trackers via one-class learning. *PETS*, 2017.
[27] M. Ikram, N. V. Rodriguez, S. Seneviratne, D. Kaafar, and V. Paxson. An analysis of the privacy and security risks of android VPN permission-enabled apps. In *ACM IMC*, 2016.
[28] J. Jeon, K. K. Micinski, J. A. Vaughan, A. Fogel, N. Reddy, J. S. Foster, and T. Millstein. Dr. Android and Mr. Hide: Fine-grained Permissions in Android Applications. In *ACM SPSM*, 2012.
[29] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *USENIX Sec.*, 2016.
[30] S. Seneviratne, H. Kolamunna, and A. Seneviratne. A Measurement Study of Tracking in Paid Mobile Applications. In *ACM WiSec*, 2015.
[31] N. Wang, B. Zhang, B. Liu, and H. Jin. Investigating effects of control and ads awareness on android users' privacy behaviors and perceptions. MobileHCI, 2015.
[32] C. Wills and D. Ununoglu. What Ad Blockers Are (and Are Not) Doing. Technical Report, 2016.