ModZoo: A Large-Scale Study of Modded Android Apps and their Markets

1st Luis A. Saavedra
Computer Laboratory
University of Cambridge
Cambridge, United Kingdom
luis.saavedra@cl.cam.ac.uk

2nd Hridoy S. Dutta
Computer Laboratory
University of Cambridge
Cambridge, United Kingdom
hridoy.dutta@cl.cam.ac.uk

3rd Alastair R. Beresford
Computer Laboratory
University of Cambridge
Cambridge, United Kingdom
alastair.beresford@cl.cam.ac.uk

4th Alice Hutchings
Computer Laboratory
University of Cambridge
Cambridge, United Kingdom
alice.hutchings@cl.cam.ac.uk

***Abstract*—We present the results of the first large-scale study into Android markets that offer modified or *modded* apps: apps whose features and functionality have been altered by a third-party. We analyse over 146k (thousand) apps obtained from 13 of the most popular modded app markets. Around 90% of them are altered in some way when compared to the official counterparts on Google Play. Modifications include games cheats, such as infinite coins or lives; mainstream apps with premium features provided for free; and apps with modified advertising identifiers or excluded ads. We find the original app developers lose significant potential revenue due to: the provision of paid for apps for free (around 5% of the apps across all markets); the free availability of premium features that require payment in the official app; and modified advertising identifiers. While some modded apps have all trackers and ads removed (3%), in general, the installation of these apps is significantly more risky for the user than the official version: modded apps are ten times more likely to be marked as malicious and often request additional permissions.**

***Index Terms*—Android, mobile apps, security, sideloading, piracy**

Android has an open design philosophy, allowing users to easily install apps outside Google Play (sideload). Thus, alternative third-party markets have emerged that allow developers to share their apps in countries where Google Play is not present (China and North Korea), or does not allow paid apps and IAPs (In-App Purchases) (Cuba, Russia and Belarus).There are open source markets such as F-Droid, and device manufacturers like Samsung, Huawei and Amazon may pre-install their own market app. *Modded apps* are defined in this study as apps with code or metadata modified by an unauthorised developer or third-party. Therefore, *modded markets* are app markets that focus on or advertise a large catalogue of predominantly modded apps. Modded apps may (for free): unlock subscription features, provide infinite in-app or in-game currency, eliminate adverts and offer paid apps. This allows users to save money or try apps, games, and subscriptions before obtaining them from legitimate sources; to enjoy an ad-free experience; and to have an advantage over others or save time on games.

Modded markets are an important part of the Android ecosystem, offering millions of apps with clear benefits and desirable features to users. However, the extent of the modifications, security implications for users, and developer and market operator incentives are less obvious and so far unstudied. We fill this gap in knowledge by identifying 423 modded markets, studying their size, presence of ads and blogs, and ranking by popularity. We then analyse over 146k (thousand) apps and their metadata obtained from the 13 most popular modded markets over a 3-month monitoring period, and match these apps with their Google Play equivalents. This allows a direct comparison between modded and official versions. The larger modded app markets operate at scale, with an average of over 37k apps (max 221k) and around 2k apps added every fortnight.

They likely reduce developers' and official markets' income. Around 5% of the apps are free copies of paid apps on Google Play, with a total value of USD $33 975 and estimated lifetime revenue in the Google Play (price $\times$ Google Play installs) of over $2 billion. Premium features usually charged via IAPs in popular apps are available for free in modded versions, e.g. ad-free audio in Spotify, which reports billions in IAP revenue per year [1]. Also, 21% of modded apps with ad IDs (advertiser IDs) have different ones to the official version in Google Play, and 6% of modded apps include additional ad libraries, potentially redirecting ad revenue away from the original developer. Modded apps are riskier for users: many modded apps claim to remove ads but only 3% do so, 23% of modded apps request additional permissions, and nearly 9% of apps are marked as malicious by VirusTotal, around 10 times the rate found in Google Play versions. Sideloading is possible in consumer laptop and desktop operating systems and gives users more choice, allowing developers to sell apps and features without paying a percentage of revenue to official markets. However, our work shows modded apps have significant negative effects. While third-party markets have the potential to benefit users and developers, some regulation is required to protect users and developer revenue streams. This work is timely for regulators, who need to balance competition and fair markets with user and intellectual property protection. The requirements to allow sideloading in the EU are being enforced and adapted by the EU's Committee on Internal Market and Consumer Protection (IMCO) [2], [3]. In summary, we make the following contributions:

- An overview of the modded app ecosystem and the first in-depth study of markets containing modded Android apps.

2159-1245/24/$31.00 ©2024 IEEE
DOI 10.1109/eCrime66200.2024.00018

- Monitoring, data collection and analysis of 13 of the most popular modded markets over a three-month period, collecting 146k modded Android apps.
- We make our dataset, *ModZoo*, available to other researchers.
- Matching modded apps with their Google Play counterparts, we find around 90% of apps are modified in some form and 75% have modified code.
- The presence of these modded markets is likely to reduce income for app developers and official markets due to: the widespread availability of paid apps for free; premium features offered for free; and the redirection of ad revenue, including 21% of apps with altered ad IDs.
- Modded apps are riskier for consumers: 23% of modded apps requested additional permissions and nearly 9% were marked as malicious by VirusTotal.

I. Research Questions and Methodology

The main research questions this paper answers are:

RQ1 What do the modded markets and apps ecosystem look like, and what is its size?

RQ2 What are the financial incentives for operating modded app markets? How does this affect the original developers and markets?

RQ3 What are the security implications of installing apps from these markets?

A. Identifying and ranking modded markets

We obtain a list of 423 sideloading and modded app markets by querying two popular search engines: Google Search and DuckDuckGo, using the following keywords in English, Chinese, Hindi and Russian: 'Android app stores', 'free Android app store', 'mod {apk/Android/games}', 'download premium apk', 'download paid apps free', '{paid/mod/premium} apps for free', 'unlocked android {apps/games}', '{YouTube/-Spotify/Trucaller} mod'. Chinese and Russian were chosen due to the limited availability of Google Play in China and Russia. Hindi was added as preliminary results included Indian domains ('.in'). We manually verified the existence of apps advertised as modded apps in the markets.

All 423 markets cannot be analysed in depth, thus a popularity-based ranking of the markets was curated using Google Trends. While only useful to compare the popularity of keywords over time, pair-wise comparisons for the 6-month period leading to our study allow us to obtain a relative ranking for all markets. We then cross referenced this ranking with the Tranco ranking corresponding to the 9-month period leading up to our study [4]. We found we analyse the top 7 most popular markets in the Tranco ranking, 9 out of the top 10, and other 4 markets within the top 35. Interestingly, out of the 423 markets, only 38 out of the top 60 in the Tranco list still offer modded apps three months later. The rest no longer operate or now focus on other activities such as offering news articles.

B. Nomenclature

The 146k *modded apps* in our study each have a unique hash and correspond to 48 384 unique package names, i.e. they are different modded versions of 48k unique apps. We refer to *exact matches* where we find an app one market with the exact same package name and version code as seen in another market. Unless stated otherwise, we use exact matches for all our comparisons. We use the *non-exact, latest-available match* when comparing a potentially malicious apps found on a modded market with the latest version of an app with the same package name on Google Play. Non-exact matches are a reasonable proxy when studying maliciousness as we assume later versions of the same app on Google Play are at worst similarly malicious to older versions. Non-exact latest-available matches are also the latest and only versions available in Google Play, so sections looking at app and IAP prices use the latest version metadata directly from Google Play as we were unable to find a reliable source of historic price data. Modded APKs and their Google Play matches are analysed and the resulting profiles are stored for later comparison. We will refer to apps on Google Play which cost money as *paid apps*, while any exact matches on modded markets are referred to as *pirated apps* because they are offered without charge on modded markets.

In later sections we discuss five different types of app. *Hash-identical* apps are those where the entire binary is hash-identical to their Google Play counterpart, i.e. where the entire packaged application (APK) is bit-for-bit identical, including manifest, libraries, code, etc. We also explore *code-identical* apps: those whose code (.dex) files are the same, but other aspects, including permissions and manifest might differ. Similarly, *certificate-identical*, *permission-identical*, *ad library-identical*, and *ad ID-identical* apps, are those whose signing certificate, permission set, ad libraries set and advertising IDs are the same as found in their Google Play version, respectively. Their counterparts are *code-modded*, *certificate-modded*, *permission-modded*, *ad library-modded*, and *ad ID-modded* apps.

C. ModZoo dataset collection

Our ModZoo dataset consists of 146 162 downloaded modded apps, their metadata and analysis results as well as their 87 792 exact and non-exact, latest-available matches from Google Play. We obtain Google Play apps from AndroZoo, a dataset which includes 21 million apps from Google Play, including different versions of the same app [5]. We scraped the 13 most popular modded markets (see §I-A) between September and December of 2022 every 10-14 days to build our dataset of modded apps. Our custom parallelised scrapers are written in Python3 to quickly obtain all relevant pages and APKs from the 13 modded markets. We used a set of proxies around the world to perform our data collection. Some of the scrapers use only HTML requests, while others also require Selenium and Mozilla's Gecko Driver to imitate user interaction. For other markets, we scraped their website first and then contacted the endpoints used by their custom

market app. All information pages were stored, including the download pages, and all available modded APKs were downloaded.

We compute SHA256 hashes of all APKs to store each app with a particular hash only once. We map modded apps to their Google Play counterparts to enable a comparison between modded APKs and their official versions found in Google Play (AndroZoo). ModZoo also includes the VirusTotal analysis results of 175 584 APKs, including 103 914 modded and 71 670 Google Play APKs. The difference between the size of our ModZoo dataset and the number of VirusTotal analysis results is due to the use of existing results, as previous studies have found VirusTotal results to be more reliable after repeated scans [6], [7].

We make the ModZoo dataset available to the research community, dataset access can be gained at https://www.cambridgecybercrime.uk/datasets.html.

D. Static analysis methodology

Static analysis allows relatively quick results, ideal for the ModZoo dataset of more than 146k modded apps and their almost 88k Google Play counterparts.

Our analysis pipeline starts by obtaining the latest data from AndroZoo. Then, it analyses the modded APKs in parallel, returning and storing their metadata and closest AndroZoo match, as well as whether it is an *exact* or *non-exact, latest-available* match (see §I-B). It then analyses the obtained AndroZoo match and stores the results. We run the third-party reverse engineering tool Apktool [8] on each app and use the UNIX 'keytool' command to obtain certificate information from each app. We obtain the certificate 'owner', 'issuer', 'serial number', 'certificate SHA256', 'signature algorithm', etc. Running Apktool again creates the 'apktool.yml' file, which we parse to obtain the APK's filename, minimum and target SDK versions, and version name and code. Our *Manifest Parser* parses the 'AndroidManifest.xml' file using a third-party Python XML library returning metadata attributes including the app's package name and version, permissions, activities, providers, receivers, intents, etc.

To detect advertising libraries in the analysed APKs and their Manifest files, a 'safelist' of ad library package names was created and iteratively extended as explained below. The *Manifest Parser* analyses the 'application', 'meta-data' and 'activity' attributes thoroughly, as this is where AppLovin and GoogleAds ad IDs, as well as the presence of IAPs can be found. We check whether the application attributes are present in our ad libraries safelist. If not in our list, it is added to a list of potential candidates to join the list, to be manually checked later. Thus, we have continuously expanded our safelist of ad libraries and reanalysed apps which analysis was older than the latest version of the safelist. All of the information gathered is stored as a profile in JSON format. The results returned to the analysis pipeline are: the package and version name, the JSON profile, ad IDs and ad libraries found, and ad library candidates. Then, the package name obtained from the manifest file and version code from the Apktool step are used to obtain from AndroZoo – where available – the *exact* or *non-exact, latest-available* match Google Play app. The AndroZoo match app analysis follows the same steps, except the AndroZoo step is skipped. The modded app analysis results are stored, as well as those of their AndroZoo match.

1) Modded apps Google Play matching: A total of 136 620 out of our 146 162 downloaded modded apps were matched with a Google Play app present in AndroZoo using the methods described above. The 6.5% unmatched modded apps correspond mostly to paid apps and games not available in AndroZoo. Out of those matched, 88.6% are exact matches (same version number and package name), and only 11.4% are non-exact, latest-available matches (same package name but the latest version number available at the time of the analysis).

E. Ethics

Our institutional ethics committee approved our study. Our ModZoo dataset contains publicly-available Android apps and their metadata, and is shared with other researchers after a thorough approval process. Apps were only collected for analysis, not used, except for the case study of 28 modded apps (see §II-B). We only used the apps long enough to test the modded functionalities, used testing devices and accounts, using no personal data. We only installed the 28 apps one at a time through markets' websites or apps. We did not undertake any activities which could affect other users, e.g. creating two accounts to look up the other user's account details in Truecaller. We contacted all market operators for comment using publicly-available contact details, stating our affiliation and purpose.

II. The modded app ecosystem

This section tackles RQ1: "What do the modded markets and apps ecosystem look like, and what is its size?" We leverage insights from our manual analysis of the 423 markets and the static analysis of our 146k app dataset obtained from the 13 most popular modded app markets.

A. Analysis of modded apps and markets

Our technical analysis focused on the 13 highest-ranked modded markets, as determined in §I-A. Their average estimated size based on number of apps listed is 37 486 apps with a mean of 15 719 apps downloaded (see Table I). The difference is due to unavailability of some apps and broken download links. Markets marked with asterisks (∗) in Table I were partially scraped as they label modded apps clearly.While this made scraping them feasible, further analysis revealed apps not labelled as 'modded', labelled 'unmodded' or 'original' are rarely hash-identical to their exact matches from Google Play, highlighting the inaccuracies of these labels. We obtained more than 10k app samples from 'Appvn', 'Androeed', and '5play' and compared them to those in AndroZoo based on their (SHA256) hashes since all signatures, metadata and code should be identical for unchanged apps. Out of the self-reported unmodded apps: Appvn had 35.8% hash-identical apps (bit-for-bit identical to Google Play apps), higher than the

0% found in the modded side of the market; Androeed had only 6.5% hash-identical apps, up from 5.7%; and 5Play 8.3%, down from 10.0% in the modded side of the markets.

Our smallest market analysed is 'An1' with 2 696 unique apps downloaded, and 'Moddroid' is the biggest with >30k. Finally, 'Appvn' has the biggest estimated size (>220k). The number of distinct apps (package names) is halved as markets often provide multiple versions of each app, unlike Google Play which only offers apps' latest version. Apps on these markets change frequently, with around 4k apps added weekly across all markets. However, around 25% are hash-identical duplicates found in more than one market. 'Duplicates' are apps advertised as different versions within a market which turn out to be hash-identical.

The 'Modded Apps' and 'Unchanged Apps' columns present the number of apps that have been modified and those that are hash-identical copies of Google Play apps, respectively. Focusing on exact matches, we compute the number of code-identical and code-modded apps as defined in §I-C: 81 250 apps (68.1%) have received changes to their code ('Modded Code' in Table I). Code-modded apps are closely related to permission-modded apps, as discussed later (see §II-C). Interestingly, although the markets focus on code-modded and modded apps, some of them have more code-identical than code-modded apps. This could be due to several reasons, the simplest being trying to offer a wider catalogue.

1) Categories and features: Modded markets focus heavily on games, we computed the Google Play categories of the modded app matches and those of a random sample of 100k Google Play apps. As shown in Figure 1, the 9 most popular modded app categories are game categories: 'Action', 'Simulation', 'Arcade', 'Puzzle', 'Casual', 'Strategy', 'Role Playing', 'Adventure' and 'Racing'. Google Play categories, however, are led by 'Education', 'Business', 'Tools', 'Health and Fitness', 'Lifestyle', 'Finance', etc. most of which are at the tail end of modded app categories.

2) Modded features: Modded markets typically provide descriptions of modded app features to inform and entice potential users. The Android catalogue has gradually shifted towards 'Freemium' apps [9]: apps with IAPs or subscriptions, and typically ads. Thus, the most popular modifications are associated with Freemium apps and games: mod money, 19 206; unlimited money, 7 202; free shopping, 3 680; original, 3 562; premium unlocked, 1 257; full version, 1 095; mod menu, 1 020; mod premium, 895; mod unlocked, 730; unlimited coins, 702; no ads, 289.

3) Paid (pirated) apps: There are 6 984 pirated apps in the 13 markets, corresponding to 2 241 package names. The price distribution in Google Play is shown in Fig. 2, showing around 40% have prices over $5 in Google Play. Roughly 48% of them have >500k Google Play installs. Their total value is USD $33 975, and $9 674 when counting each app (package name) only once. Their approximated lifetime revenue in Google Play (US price × number of global installs in Google Play) is $2.28 billion. All modded markets studied lack payment mechanisms, thus the paid apps in Table I are available for free, and likely pirated copies of paid Google Play apps. The mean percentage of paid apps available for free across all markets is 4.7%, with only 'Malavida' hosting 0.0% (6 apps in total).We estimate modded market operators would have spent at least $9 674 to get these paid apps from Google Play before hosting them in their markets if they had worked together and shared all their apps, or $26 901 if they had to buy each of the paid apps they host once. It is possible market operators downloaded paid apps and requested refunds after making copies [10], resulting in $0 of revenue for the original developers.

4) In-App purchases (IAPs): The 13 modded markets contain 100 118 apps with IAPs worth at least $3.7 million, with prices of up to $1 024 per item in Google Play. Over 40% of the apps have IAPs with prices over $100 in Google Play. Their total price is at least $3.7 million, based on the values reported on Google Play. Much IAP content and features are free in modded apps (see §II-B). The maximum, minimum and mean IAPs prices (in Google Play) are shown in Fig. 2.

5) Countermeasures and market changes: During data collection we encountered different countermeasures employed by markets against scraping and automated downloads. We observed that all markets analysed contain duplicate apps found in others. Waiting periods are common, preventing users from downloading multiple APKs in a short period, it provides an opportunity to show users more ads. Some markets used CAPTCHA tests, and a small number implemented Cloudflare DDoS and bot protection [11]. Some markets introduced anti-scraping protections during our study. It is possible that our contacting markets for comment, scraping activity, or both made operators suspicious and more security-conscious. One market removed their 14 social media links (Github, LinkedIn, YouTube, etc.) from the English but not the Vietnamese version of their site during our study.Some markets require installing a proprietary app to download APKs they host, e.g. Moddroid, Jojoy, and HappyMod, which host mostly the same APKs in a shared back-end. Our scrapers obtain the metadata from the websites and contact the shared endpoints to download the APKs directly.

B. Case study

The five all-time most popular apps and games (as of March 2023) from Google Play are presented as a case study. We manually test the official alongside modded versions from different markets to analyse their modded features and assess the scale of revenue loss caused by modded apps. Google Play Protect is supposed to warn users of harmful apps on their devices, even when sideloaded. It may also deactivate or remove harmful apps. During our case study it warned "Unsafe app blocked" for 2 out of the 30 modded apps (28 plus 2 market apps), these were a game and the 'Apkmody' market app. Users can click "Install anyway".

Many modded apps showed a small logo or pop-up with the market's and sometimes modder's name. In some cases the market name displayed differed from the market we obtained the app from. We found 14 out of the 28 app pairs studied

TABLE I
OVERVIEW OF THE MODDED MARKET ECOSYSTEM AND PROPORTION OF MODDED APPS.

Market	Estimated Size	Unique Apps	Unique Packages	Dupli-cates	Paid (%)	Modded Apps	Unchanged Apps	Modded Code
Appvn∗	221 039	∗4 389	∗1 866	∗8	∗5.0	∗4 297	∗0	∗3 586
RevDl	42 540	30 477	9 599	187	6.4	23 217	3 466	5 068
HappyMod	41 385	26 737	17 249	12	3.7	19 996	4 098	12 826
MODDROID	34 312	30 738	17 152	13	3.7	23 316	4 005	15 153
APKMODY	33 914	10 516	3 081	214	4.5	8 857	420	4 550
androeed∗	24 252	∗15 450	∗6 869	∗6 195	∗3.4	∗12 069	∗731	∗9 163
Rexdl	22 988	14 262	5 824	24	8.4	11 666	1 822	2 621
5play∗	19 014	∗19 674	∗15 859	∗16 203	∗8.1	∗16 095	∗1 610	∗9 917
Malavida	16 519	19 648	16 128	16	0.0	14 333	4 115	3 084
APKDONE	11 080	14 908	3 232	113	4.3	10 099	139	7 341
ApkVision	8 491	7 983	6 900	16	5.9	5 632	1 055	2 683
LMHMOD	7 880	6 865	4 317	6 303	3.7	5 577	229	3 597
An1	3 906	2 696	1 198	44	4.0	2 629	22	1 661

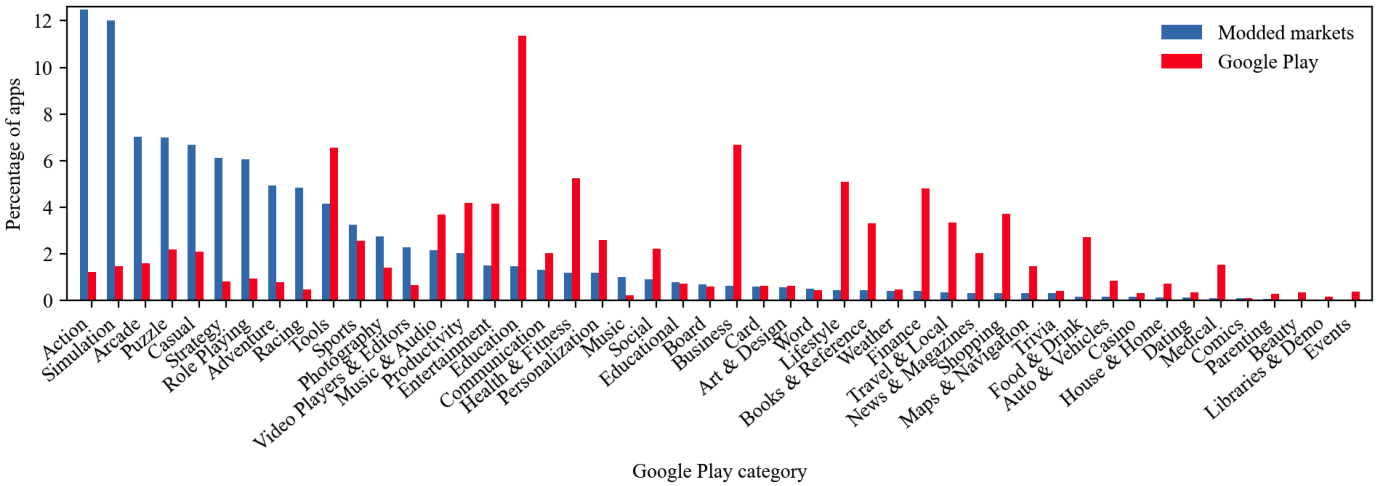

Fig. 1. Distribution of Google Play app categories in the modded markets and Google Play

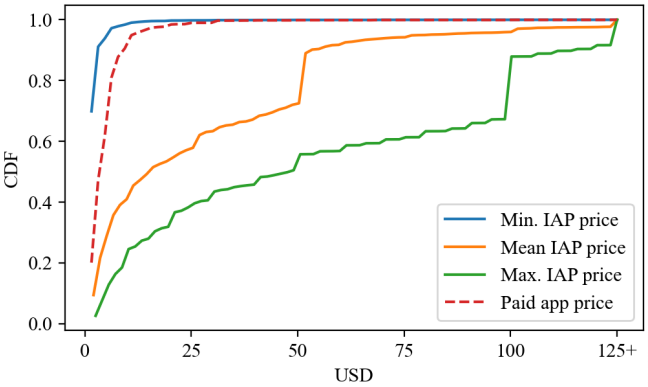

Fig. 2. Google Play paid apps and IAPs price CDF.

had Google Mobile Ads and/or AppLovin ad IDs present. Of these, one TikTok and one Truecaller version had their Google Mobile Ads IDs removed.

1) TikTok: reported $1.5 billion IAP revenue in 2022 [12], these are coins users can send to creators during livestreams, resulting in revenue for creators. They cost USD $0.07–249.00 for 5–17 500 coins. Modded versions claimed to offer unlimited coins, downloads without watermarks and no geolocation restrictions. Downloads worked well but coins were not included in any versions we tried.

2) SHAREit: premium features for $1.99/month include no ads, exclusive customer service, regular cleanup and antivirus. Modded versions claim to remove all ads and include all premium features. None of the apps tested was ad-free, one provided regular cleanups and none provided premium customer service.

3) Telegram: Premium for $4.99/month, or $35.99/year offers no ads and doubled limits (channel size, download speeds, document size, etc). Modded versions claim to provide these. Ads are only shown in public channels and were not served to us in genuine nor modded versions. Modded apps tested did not provide any other features, with download speeds as limited as free versions.

4) Spotify: Premium for $9.99/month offers no ads, higher sound quality, playing songs in any order, unlimited skips, downloads and offline listening. Spotify reported 2 million

users ran modded versions in 2017 to avoid audio ads and subscriptions [13]. They reported a €11.57 billion revenue from Premium subscriptions and €1.7 billion ad revenue from non-premium users in 2023 [1]. Modded versions advertise having the premium features. They were ad-free, provided unlimited skips and the ability to play any song. Not all make it clear they cannot provide downloads, offline listening and high quality audio.

5) Truecaller: Premium for $4.99/month or $49.99/year offers no ads, advanced spam blocking, seeing who viewed your profile, incognito mode, etc. Modded versions claim to have all premium features. However, although ad-free, the modded versions tested show all users as Gold members, with no effect for genuine users. Only one version showed who viewed or searched the user's profile.

6) Subway Surfers: 'coins' and 'keys' bundles cost $0.99–99.99. Modded versions claim to have all these IAPs unlocked, some offer 'God mode' game-play advantages: unlimited jumps, flying, etc. Piracy cost this game $91 million by 2017 [14]. The versions we tested provided free IAPs and unlimited coins.

7) Candy Crush Saga: offers many perks from $0.99–99.99. All modded markets advertise having all levels unlocked, infinite lives, boosters, etc. Such offerings render all IAPs useless. The modded versions tested worked as advertised.

8) Free Fire: 'diamonds' cost $0.99–49.99, perks $8–12.99/month. Modded markets advertise diamonds and gameplay-related mods: aim-assist, no recoil, etc. Most versions tested did not work and the rest had none of these features.

9) My Talking Tom: 'diamonds' cost $1.99–99.99, perks $4.99/month. Purchases remove all ads. Modded apps offer unlimited coins, perks and no ads. Unlimited coins unlock most content, but some ads and locked perks remain.

10) Hill Climb Racing: perks and coins cost $1.99–59.99, some remove ads. Modded versions advertise unlimited coins or all content unlocked. All versions tested provide unlimited coins, unlocking all content, although ads remained.

C. Code, permissions and ad libraries

We study changes to the code, stored in the 'classes[n].dex' file(s) in relation to changed permission sets, ad libraries, and ad IDs for all exact matches. Of these pairs, a majority (75.0%) are code-modded, 38.4% of them are also permission-modded, with the majority (59.4%) including additional permissions (as shown in Fig. 3). Some code-modded apps require further permissions for reasons related to the modifications, but the reason behind many additions was unclear. Code-modded apps are mainly ad library-identical (83.3%), only 11.1% of them have fewer. In terms of ad IDs, 36.2% of code-modded apps had none, and of those with ad IDs, 21.6% had them changed. Altered ad IDs occur in a significant proportion of code-modded apps, we hypothesise permissions are sometimes added to code-modded apps in order to increase ad revenue for the modder. For code-identical pairs, permissions and ad libraries remained completely unchanged in 99.9%, and 100% of the pairs, respectively. Their ad IDs either remained unchanged (65.3%) or no ad ID was found. This suggests many apps are copied directly from Google Play without alteration, maybe to expand the catalogue and engage users even if a mod is unavailable. Many markets offer modded and original versions of each app, although we found inaccuracies in these labels (see §II-A). Unmodified apps may be useful to users for compatibility or features.

D. App signing certificates

Most markets use mainly debugging and default Android Studio certificates unfit for app publishing. Some market-specific signatures such as '5play', which signs most code-modded apps with it. So does Apkmody, which includes the operator's name 'Anh Pham' but mainly uses default signatures. Others use mainly 'A1 Lazyland RU', present in most markets. Appvn uses mainly 5play.ru, all markets have 5play.ru and/or Apkmody certificates except Malavida. All markets have a small proportion of third-party markets' and modders' certificates which include websites and Telegram links. Some were signed with 'AntiLVL'. This analysis confirmed our previous findings of cross-market duplicate apps.

This section looked at pirated apps, IAPs and a case study of popular apps and games. Most popular categories and modded features relate to games. Many apps are code-modded, bypassing subscription features or in some cases removing ads. Others added permissions, ad libraries, and changed ad IDs. 'Unmodded' labels and 'ad-free' app descriptions cannot be trusted.

III. Market operator motivations and income

This section tackles RQ2: "What are the financial incentives for operating modded app markets? How does this affect the original developers and markets?" We approach these based on our observations and analysis results to analyse possible revenue streams in modded markets and operators' economic incentives.

A. Blogs, sponsored posts and ads

We manually studied blogs, sponsored posts and ads in the 423 modded markets. Blogs are present in a third of them. They host articles about modded app updates, installation guides, and often also news articles, tips and tricks, rankings, and product or app reviews. Many markets openly displayed their pricing for advertising through different ads, product reviews, or guest posts. Others were open to contact and only a minority did not accept sponsored posts or ads. Some blogs are inactive and 13% have 5 or fewer posts. One market priced sponsored posts and ads at USD $250–300; others for $100. Most had lower prices starting at around $30 for general posts, $45 for casino-related posts, and more for those related to "gambling, adult, dating, vaping, CBD, or cannabis". Most posts are admin-uploaded, making it difficult to count the

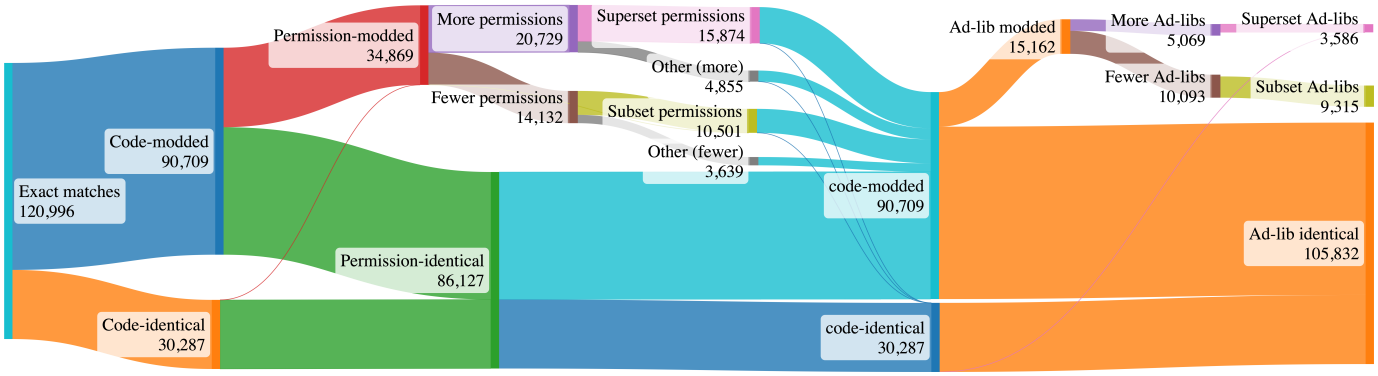

Fig. 3. Permissions and ad library changes in code-identical and code-modded apps.

sponsored posts. Some offered different ad types including sidebar and pop-ups for $50–200/month.

B. Advertising libraries and advertiser IDs

Many code-modded apps are advertised as ad-free versions of 'Freemium' apps. We use a safelist (see §I-D) to confirm whether they are ad-free and what other changes they have. Google Mobile Ads and AppLovin ad libraries (two of the most popular) include their ad ID in the manifest file, allowing us to compare them between modded and original apps. We found 20.5% of modded apps with ad IDs had them altered. It appears to be widespread practice to redirect ad revenue from the original developer to the modders or modded markets. Also, 41 321 apps use ad libraries other than Google Mobile Ads and AppLovin, and in total 10 990 apps have changed ad libraries compared to their Google Play version.

The most popular advertising and tracker libraries present in our modded apps are GoogleAds, Facebook, and Unity, followed by AppLovin, ironSource, Vungle, AdColony, Tapjoy and InMobi. Their relative popularity is mostly the same in modded apps and their Google Play counterparts. Providers most affected by 'ad-free' modded apps are GoogleAds (20.3%), Facebook (14.9%), Unity3D (9.6%), and AppLovin (8.1%). We found 10 353 contained no ad libraries originally, 4 180 (2.86%) had all removed and 2 636 had their AppLovin and GoogleAds ad IDs removed. So, while some modded apps have had ad libraries removed, they are in the minority. The presence of libraries implies the possibility of ads in an app, e.g. the popular Unity library used in many games ('com.unity3d') can be used to display ads. Thus, we may have underestimated the number of ad-free apps. Further dynamic, manual analysis would be needed to confirm this, which would be impractical given the scale of our dataset.

C. Advertising libraries, advertiser IDs and permissions

Changes to permissions have security implications. Combined with the aspects already presented in relation to ad libraries and ad IDs, they might provide increased revenue to modders or market operators. E.g. an ad library might use added location permissions to display more relevant ads. The small proportion of ad-free apps (2.86%), those which contain no ads where their Google Play counterparts do, typically present fewer permissions (88.9%), a strict subset of the original permissions (76.2%), and only 4.2% are permission-identical, as shown in Fig. 4. Thus, there is a genuine small offering of ad-free versions of popular apps with smaller permissions sets. Furthermore, as shown in Fig. 4, 91.0% of modded apps are ad-library-identical to their Google Play counterparts. These tend to be permission-identical apps (84.3%), with the rest mostly having more (11.4%) and a superset (10.3%) of permissions. Apps with added ad libraries are mainly permission-modded apps (91.3%), with 64.9% of them having more permissions. Those with removed ad libraries are mostly permission-modded (88.6%), mainly with fewer permissions (60.9%) or a strict subset (44.1%). These results show ad libraries are not typically changed and are closely linked to permissions. When focusing on their GoogleAds and AppLovin ad IDs, 58.0% of modded apps with ad libraries have unchanged ad IDs, 8.1% had changed ad IDs, and 33.9% of the modded apps had no ad IDs. Ad-ID-identical apps were mostly permission-identical (83.6%), with another 13.6% having more permissions, as shown in Fig. 5. Ad-ID-modded apps, however, were permission-modded in 61.1% of the cases, with an even split of more and fewer permissions. Those with no ad IDs were mainly permission-identical (76.7%). This suggests again a strong correlation between permission-modded, ad-library- and ad-ID-modded apps. This could be due to modders wanting to be compensated with ad revenue or more permissions giving more granular data to ad libraries, increasing ad revenue.

D. Modded apps and user displacement

Piracy has been found to have a negative impact on revenue in the music and media industries [15], [16]. Tapcore estimated 14 billion app installs were pirated in 2017, costing app developers more than $17.5 billion, and Subway Surfers $91 million [14]. It is difficult to estimate the current revenue loss with the growth of IAPs and ad revenue in apps and games since 2017. Studies on computer game piracy found high displacement rates of -2.49 for games, meaning each illegal download of a game typically displaces multiple genuine purchases [17]. Unlike for music, films, series, and books,

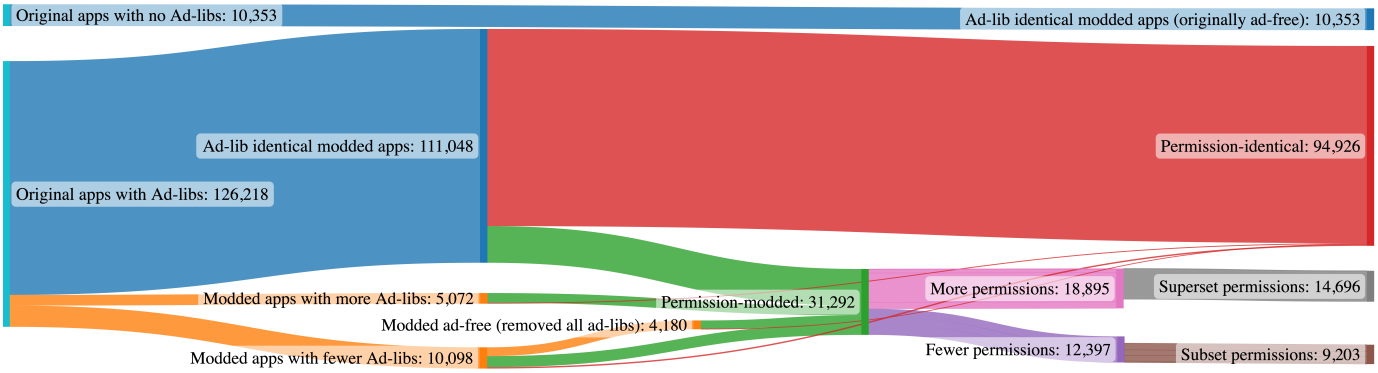

Fig. 4. Distribution of ad libraries in original and modded apps and modded permissions.

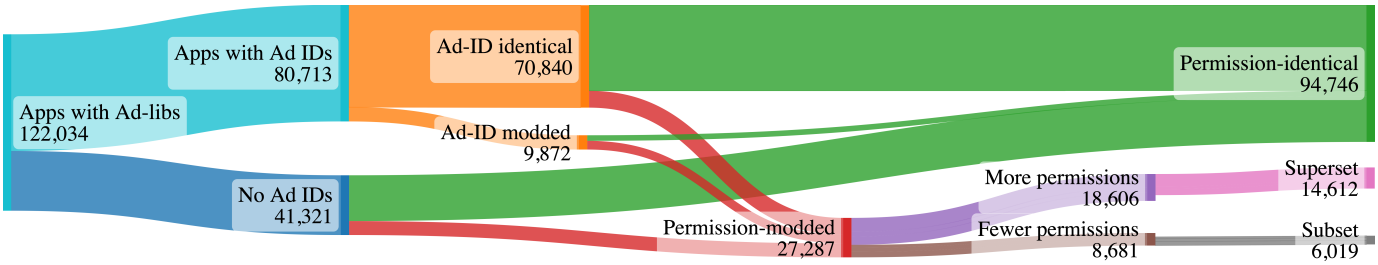

Fig. 5. Distribution of permissions and advertiser IDs in modded apps with ad libraries.

where pirates tend to increase legal consumption while decreasing illegal consumption, they found game pirates increase or maintain illegal consumption over time, resulting in user displacement and lost revenue.

We identified possible revenue streams for market operators and modders including ads and sponsored posts. Our analysis shows a correlation between ad-library-modded and permission-modded apps. Code-modded apps are typically permission-modded and some ad-library-modded. Code-identical apps are permission and ad-library-identical. Furthermore, 1 in 5 code-modded apps with ad IDs had them changed. We found 6 984 pirated apps with $2.28 billion in estimated lifetime revenue in Google Play, and 100k apps with IAPs typically offered for free in modded apps (see §II-A3–§II-B). Modded apps may cause user displacement, disrupting original developers' and markets' revenue and innovation.

IV. Security implications of modded apps and markets

This section answers RQ3: "What are the security implications of installing apps from these markets?" App analysis and VirusTotal malware analysis results are combined to tackle this from the consumers' perspective. However, there are also security and economic implications for the original app developers. Many of the modded apps use the original API keys so original developers pay for cloud services and API calls, etc. (see §II-B). Also, other users' security might be affected. Modded apps can change what users can or cannot see in social networking apps, potentially exposing other users' information beyond their preferences. It may also affect other Internet users, e.g. if modded apps embed a botnet.

A. *VirusTotal analysis*

The VirusTotal analysis methodology is based on previous approaches, VirusTotal is queried with the hashes present in the entirety of ModZoo obtaining all existing analysis results. Analysis results obtained after repeated scans have been found by previous studies to be more reliable than new results [6], [7]. Similarly, the recommended threshold of around 10% of antivirus engines (AVs) flagging APKs as malicious is used (see §VI). Furthermore, existing [18] and custom tools were used to obtain unified malicious labels. We use VirusTotal to get insights into the entire ModZoo dataset. More advanced techniques could be used on a random or selected sample of the dataset, but that is considered future work.

Modded apps are sometimes paired with the non-exact, latest-available matches when the exact version of the app is not in AndroZoo. This is reasonable since the latest-available version on Google Play should be just as safe or safer than older versions. AndroZoo has most app versions, its authors have mitigations for robust scraping [5].

1) Malware, adware, and PUPs: The VirusTotal results cover 103 914 of the modded apps from our ModZoo dataset and 71 670 Google Play (AndroZoo) apps. We found almost 9% of code-modded apps and only 0.5% code-identical apps coming from modded markets were labelled malicious compared to only 0.9% of their currently-available Google Play counterparts, as shown in Figure 6. Users are more vulnerable

to malware, adware, potentially unwanted programs (PUPs) and other malicious programs when downloading modded apps.

In total, 167 273 apps were marked as undetected and 8 311 (4.7%) as malicious. Of these, 85.3% came from modded markets and the rest from Google Play (AndroZoo). This translates as 6.82% of modded apps and 1.70% of Google Play apps in ModZoo classified as malicious. However, 8.59% of code-modded apps are malicious, against only 0.51% of code-identical apps. The risk of modded apps goes beyond this, since many of the apps offered in modded markets are no longer offered in Google Play: 13.36% of the Google Play counterparts are no longer available as of March 2023. Of these, 6.72% are marked as malicious, compared to only 0.93% of those still available. For hash-identical apps the risk is obviously identical between Google Play and modded markets while both host the app. However, Google analyses apps and responds quicker to incidents and user reports, providing better security protections than modded markets.

Malicious apps are flagged as PUPs such as LuckyPatcher, used to modify Android apps, and many are flagged with more worrying Trojan-like malware such as Andreed, Triada, RemoteCode, HiddenAds, Kyvu, (LuckyPatcher) IBGV, etc. and more general labels as 'downloader', 'virus'. The 20 most prominent labels are shown in Table II, 4 of them are not present in Google Play apps at all, while most others have a significantly bigger presence in modded apps.

TABLE II
MOST COMMON VIRUSTOTAL THREAT LABELS AND THEIR DISTRIBUTION

Threat Label	Total	Modded Apps	Google Play
None	163 788	93 541	70 247
andreed	3 830	3 761	69
grayware	3 586	3 052	534
fyben	2 111	2 078	33
adware	1 011	490	521
downloader	465	413	52
androeed	183	181	2
kyvu	72	19	53
grayware:tool	51	37	14
triada	28	7	21
remotecode	27	24	10
dataeye	24	14	10
hiddenads	23	18	5
luckypatcher	21	21	0
ibgv	18	18	0
spyware	18	16	2
tencentprotect	18	10	8
fleeceware	17	8	9
boogr	14	13	1
wamod	14	14	0
virus	13	13	0

2) Permissions in code-modded apps: Many dangerous permissions are added to code-modded apps. Malicious code-modded apps have a higher incidence of these as shown in Table III, with 14.6% adding 'SYSTEM_ALERT_WINDOW' which allows creating windows on top of any other app, which can be used for phishing attacks. A further 9.0% malicious code-modded apps added the 'READ_EXTERNAL_STORAGE' permission, which allows access to other apps' files in the MediaStore, potentially exposing users' personal data. Malicious code-modded apps are twice as likely to request these, although there might be genuine need for some of them in modded apps. The following permissions are more than 4 times more likely to be used in malicious than non-malicious code-modded apps: 'WRITE_SETTINGS' which allows apps to read system settings, 'READ_LOGS' which is not to be used by third-party apps since it allows apps to "read the low-level system log files", which may contain users' private information, and 'CAMERA'. Other risky permissions such as 'ACCESS_COARSE_LOCATION' and 'ACCESS_FINE_LOCATION' are less common but are 8 times more likely to be added to malicious apps. Google distinguishes different permission protection level categories: 'dangerous', 'normal', and 'signature'. [1] It is worth noting that although 'normal' permissions do not require user confirmation in-app like 'dangerous' permissions, they are still potentially dangerous as users of modded markets are not presented with accurate information of the 'normal' or 'dangerous' permissions used by apps. They are all android.permission.{} except 'net.dinglisch.android.tasker.PERMISSION_RUN_TASKS' and 'com.android.launcher.permission.INSTALL_SHORTCUT'. We classified them as dangerous and normal, respectively, although they are not included in Google's classifications.

The security of modded markets is significantly lower than that of Google Play, with 8.6% of code-modded apps and 6.8% of apps overall flagged as malicious by VirusTotal, against 0.9% of currently-available Google Play apps. Furthermore, we found a high number of dangerous and risky permissions in code-modded apps, especially those classified as malicious.

V. LIMITATIONS

The analysis of ad libraries in modded apps is potentially biased, as the accuracy of results depends on that of our safelist. Thus, we expanded it periodically as more apps were analysed. Code obfuscation and shrinking are techniques available to developers to make apps more secure, difficult to reverse engineer, and storage efficient. However, they also undermine static analysis of ad libraries. Ad IDs, permissions, and other parameters studied are not affected by this limitation. Several analysis tools have been proposed to study the libraries present in obfuscated apps [19], [20], They are infeasible considering the scale of ModZoo as they require downloading all libraries of interest. A safelist is an acceptable compromise between accuracy, speed and scale since code obfuscation and shrinking are not enabled by default in Android Studio. Google Play reports IAP prices as ranges instead of their number, price and which are subscriptions.

[1] https://developer.android.com/reference/android/Manifest.permission

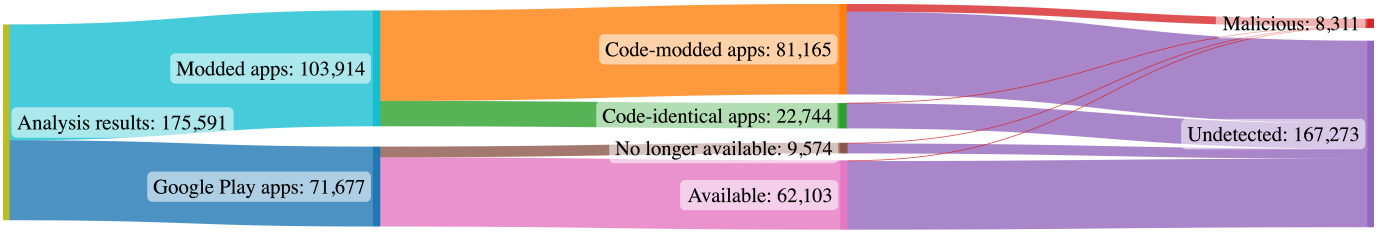

Fig. 6. Distribution of malicious apps across Google Play and modded apps.

TABLE III
TOP 30 ADDED PERMISSIONS IN MALICIOUS CODE-MODDED AND CODE-MODDED APPS AND THEIR CATEGORY.

Permission	Category	Malicious Code-Modded (%)	Code-Modded (%)
android.permission.SYSTEM_ALERT_WINDOW	signature	14.60	8.72
android.permission.READ_EXTERNAL_STORAGE	dangerous	9.01	4.10
android.permission.BLUETOOTH_ADMIN	normal	7.39	1.65
android.permission.BLUETOOTH	normal	7.19	1.62
android.permission.WRITE_SETTINGS	signature	7.05	1.70
android.permission.CHANGE_WIFI_STATE	normal	6.68	1.55
android.permission.FLASHLIGHT	normal	6.61	1.56
android.permission.USE_FINGERPRINT	normal	6.59	1.57
android.permission.READ_LOGS	very dangerous	6.50	1.54
android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS	normal	6.50	1.56
android.permission.READ_SETTINGS	uncategorised	6.50	1.56
net.dinglisch.android.tasker.PERMISSION_RUN_TASKS	dangerous	6.50	1.56
android.permission.CAMERA	dangerous	6.38	1.54
android.permission.REQUEST_INSTALL_PACKAGES	signature	5.62	1.07
android.permission.VIBRATE	normal	4.38	0.88
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	3.82	3.01
android.permission.ACCESS_WIFI_STATE	normal	3.41	0.86
android.permission.QUERY_ALL_PACKAGES	normal	2.81	6.66
android.permission.GET_TASKS	normal	1.96	0.74
android.permission.READ_PHONE_STATE	dangerous	1.38	0.63
android.permission.RESTART_PACKAGES	deprecated	1.06	0.13
android.permission.KILL_BACKGROUND_PROCESSES	normal	1.01	0.11
android.permission.RECEIVE_BOOT_COMPLETED	normal	0.90	0.15
android.permission.CHANGE_NETWORK_STATE	normal	0.85	0.10
android.permission.BATTERY_STATS	signature	0.78	0.09
android.permission.ACCESS_COARSE_LOCATION	dangerous	0.76	0.09
android.permission.BROADCAST_STICKY	normal	0.76	0.07
android.permission.ACCESS_FINE_LOCATION	dangerous	0.67	0.09
com.android.launcher.INSTALL_SHORTCUT	normal	0.67	0.09
android.permission.DOWNLOAD_WITHOUT_NOTIFICATION	normal	0.53	0.18

VI. RELATED WORK

Wang et al. analysed 6 million Android apps in 16 Chinese markets, inter-market similarity, publishing behaviours, malicious and fake apps [21]. The markets performed substantially worse than Google Play. We found similar in modded market security and presence of pirated apps. Also, we explored operator and modder motivations and revenue streams. Others studied Android app attribution, and found the lack of metadata in AndroZoo a limitation to study app attribution at scale [22]. Metadata for removed Google Play apps is lost, thus we stored modded market app metadata in our ModZoo dataset.

Other studies focused on Android VPN [23] and firmware over-the-air [24] apps, analysing their security, permissions and presence of malware through VirusTotal.

Others found free games in Google Play have 3.4 times more trackers and twice the number of dangerous permissions as paid ones [25]. Kumar et al. analysed differences in 26 countries' Google Play markets, finding apps are often unavailable due to developer-introduced geoblocking [26]. These aspects could motivate modders and users.

Shen et al. found malicious apps last more than twice as long on Google Play than manufacturer-provided markets [27]. We found the opposite for modded markets, as operators lack the motivations device manufacturers have to keep their platforms secure. Our study is also novel in the mapping of third-party (modded apps) with their Google Play counterparts to compare ad libraries, permissions sets and security implications. Others found repackaged apps aimed at tricking users to think they are genuine apps are common in official markets and half of them contained adware [28]. Instead, modded apps are advertised as modified versions. They found half of the 15k repackaged apps contained adware, against our 9% malicious code-modded apps. However, only 4% of

them added permissions against 24% of code-modded apps in our study. They did not study ad IDs and their results are not reproducible due to their dataset's unavailability.

Previous research separated prominent and trivial permissions [29], created permissions graphs to find outliers [30], [31], and found malware-related permissions based on other datasets [32]. They rely on existing datasets or do not share their own. Unlike ours, they do not consider the connections between ad libraries and permissions changes. We also explored permissions added to malicious code-modded apps and found increased use of dangerous permissions. Others studied manifest file intents and context [33], [34], while some identified packages and APIs used [35]. Static analysis is common to these large-scale approaches. We combine it with market and VirusTotal analysis.

Previous studies have used VirusTotal to analyse apps at scale. Zhu et al. surveyed 115 papers to identify VirusTotal analysis methodologies [36]. They collected executables analysis results for a year and found the threshold approach (labelling files malicious when flagged by at least N antivirus engines included in VirusTotal) outperforms others. Most papers use thresholds to classify malicious files and the most popular threshold, $t = 1$, does not perform well [36]. They recommend small thresholds $>$1, such as 2 to 15; we used a 10% (5–7) threshold. Others analysed a small sample of 9k apps from 9 third-party Android markets with a threshold of 6 and found 5% apps were malicious [37]. 31% had not been analysed by VirusTotal, yielding no results. Furthermore, we found a higher proportion of malicious apps in modded markets, and compared modded apps to their Google Play counterparts. Most approaches use a similar approach with different thresholds [38]. Others used weighted voting, relied on supervised learning, and used future results (after 4 weeks) as ground truth [39]. Others confirmed the increased accuracy of older results [6], [40]. Others focused on native code libraries misuse in a small sample of Android apps from one third-party market [41]. It required manual verification for some types of misuse, unsuitable at scale. Similarly, others identified harmful libraries in Android and iOS based on VirusTotal results [42]. Our study links the presence and changes of ad libraries with changes in ad IDs and permissions.

Previous research has explored sideloading user motivations and knowledge [43], but they did not consider modded apps nor motivations of the maintainers. Their questionnaire is run on a sample of Computer Science students and staff, as well as relevant sideloading and rooting Reddit forums users, thus providing limited data on the real-world occurrence of sideloading.

VII. Conclusion

This paper presented the results of the first large-scale study into modded app markets, with a large-scale technical analysis of 146k modded apps available on the 13 most popular markets. By comparing them to their Google Play counterparts, we demonstrated the vast majority were modified in one or more ways, including those labelled as unmodified. Our updated dataset with 300k apps is publicly available. Modded markets likely reduce developers' and official markets' income due to pirated apps and unlocked premium features. The majority of apps fell into the gaming category, however many other popular apps exist on these markets, including a modified version of TikTok advertised as offering free coins and a modified version of Spotify offering ad-free music without subscription. Modded apps add permissions and ad libraries; 21% had different ad IDs than their Google Play version, suggesting ad revenue may be diverted away from original developers.

From the users' perspective, modded apps advertise new, desirable features, which our case studies show often, although not always, work. However, these markets are unrelated to the genuine developers, and divert or curtail the app purchase, IAPs and ads revenue streams. While ad-free versions of apps are widely touted, fewer than 3% of modded apps had all ads and trackers removed. VirusTotal marked 9% of code-modded apps as containing adware, grayware or Trojans, 10 times the rate found in Google Play. Modded markets continue to host malicious apps removed from Google Play for a long time. Users risk their personal data and their privacy being breached by downloading apps which code has not been verified by any official entity and modified by third-parties. Furthermore, users might put others' privacy and security at risk, as modded apps might allow private content to be viewable by others, malware and spyware might make use of added permissions to access other users' private information, contacts, etc.

Developers should be aware of these markets and practices, and given more tools and support to find and report malicious versions of their apps. Developers, especially smaller ones, will have a hard time reporting misuse of their intellectual property since at present they would need to manually report multiple versions of their apps in more than 400 modded Android markets. With current legislation such as DMCA, this only covers those versions flagged, so with every app update appearing in the markets they would have to repeat the process.

The question of whether and how mobile devices should allow installing apps outside the official market is under investigation by regulators in the EU and the UK. Android offers official support, while iOS makes it very hard for the average consumer to install apps from outside the official market. Our work suggests regulators should consider options to counter the negative effects of modded markets and sideloading while protecting or enhancing user and app developer choice and protection. There are a range of options, including allowing sideloading while requiring apps to be tested and signed by an approved tester; requiring the distribution of alternative market apps through the official market in order to offer a pinch-point to support regulation; etc. A confounding factor is that large revenue streams are tied to the status quo where a percentage of the price of paid apps and IAPs flow to the official market operator.

ACKNOWLEDGMENT

We thank Richard Clayton for his help in setting up and maintaining the scraping machines, proxies and storage. We thank Stan (Jiexin) Zhang for his contributions to some scrapers. We are grateful to VirusTotal for the academic access to their API. We thank colleagues and anonymous reviewers for their feedback. This work is supported by Nokia Bell Labs (for LAS and ARB) and the European Research Council under the Horizon 2020 programme (grant agreement No 949127) (for HSD and AH).

REFERENCES

[1] M. C. Götting, "Spotify's revenues from 2012 to 2023 by segment," Feb 2024, https://www.statista.com/statistics/245125/.

[2] "Europa.eu", "2020/0374(COD) Digital Markets Act," 2021.

[3] ——, "Deal on Digital Markets Act: EU rules to ensure fair competition and more choice for users," 2022, https://www.europarl.europa.eu/news/en/press-room/20220315IPR25504.

[4] V. L. Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," *arXiv preprint arXiv:1806.01156*, 2018.

[5] K. Allix, T. F. Bissyandé, J. Klein, and Y. Le Traon, "Androzoo: Collecting millions of android apps for the research community," in *Proceedings of the 13th international conference on mining software repositories*, 2016, pp. 468–471.

[6] A. Salem, S. Banescu, and A. Pretschner, "Maat: Automatically analyzing VirusTotal for accurate labeling and effective malware detection," *ACM Trans. Priv. Secur.*, vol. 24, no. 4, jul 2021. [Online]. Available: https://doi.org/10.1145/3465361

[7] H. Wang, J. Si, H. Li, and Y. Guo, "RmvDroid: Towards A Reliable Android Malware Dataset with App Metadata," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, 2019, pp. 404–408.

[8] iBotPeaches, "Apktool," 2023, https://github.com/iBotPeaches/Apktool.

[9] S. Hill, "Freemium apps: necessary evil or plain greedy?" *Android Authority*, 05 2014, https://www.androidauthority.com/freemium-model-good-bad-thing-384124/.

[10] "Google.com", "Learn about refunds on Google Play," 2023, https://support.google.com/googleplay/answer/2479637.

[11] "Cloudflare.com", "Cloudflare DDoS Protection & Mitigation," 2023, https://www.cloudflare.com/en-gb/ddos/.

[12] J. Koetsier, "TikTok earned $205 million more than Facebook, Twitter, snap and Instagram combined on in-app purchases in 2023," Mar 2023, https://www.forbes.com/sites/johnkoetsier/2023/03/01/tiktok-earned-205-million-more-than-facebook-twitter-snap-and-instagram-combined-on-in-app-purchases-in-2023/.

[13] C. Gartenberg, "Spotify reveals 2 million free users are dodging ads," Mar 2018, https://www.theverge.com/2018/3/23/17156014/spotify-users-premium-modded-hacked-app-free-streaming-music.

[14] J. Koetsier, "App developers losing $3-4 billion annually thanks to 14 billion pirated apps," Jul 2017, https://www.forbes.com/sites/johnkoetsier/2017/07/24/app-developers-losing-3-4-billion-annually-thanks-to-14-billion-pirated-apps/.

[15] R. Rob and J. Waldfogel, "Piracy on the high C's: Music downloading, sales displacement, and social welfare in a sample of college students," *The Journal of Law and Economics*, vol. 49, no. 1, pp. 29–62, 2006, https://doi.org/10.3386/w10874.

[16] M. D. Smith and R. Telang, "Assessing the academic literature regarding the impact of media piracy on sales," *SSRN Electronic Journal*, 2012.

[17] J. Poort, J. Quintais, M. A. van der Ende, A. Yagafarova, and M. Hageraats, "Global Online Piracy Study," *Amsterdam Law School Research Paper*, no. 2018-21, 2018.

[18] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "AVclass: A Tool for Massive Malware Labeling," in *Research in Attacks, Intrusions, and Defenses*, F. Monrose, M. Dacier, G. Blanc, and J. Garcia-Alfaro, Eds. Cham: Springer International Publishing, 2016, pp. 230–253.

[19] J. Zhang, A. R. Beresford, and S. A. Kollmann, "Libid: reliable identification of obfuscated third-party android libraries," in *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2019, pp. 55–65.

[20] Y. Wang, H. Wu, H. Zhang, and A. Rountev, "Orlis: Obfuscation-resilient library detection for android," in *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*, 2018, pp. 13–23.

[21] 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 Internet Measurement Conference 2018*, 2018, pp. 293–307.

[22] K. Hageman, Á. Feal, J. Gamba, A. Girish, J. Bleier, M. Lindorfer, J. Tapiador, and N. Vallina-Rodriguez, "Mixed signals: Analyzing software attribution challenges in the android ecosystem," *IEEE Transactions on Software Engineering*, 2023.

[23] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. A. Kaafar, and V. Paxson, "An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps," in *Proceedings of the 2016 internet measurement conference*, 2016, pp. 349–364, https://doi.org/10.1145/2987443.2987471.

[24] E. Blázquez, S. Pastrana, Á. Feal, J. Gamba, P. Kotzias, N. Vallina-Rodriguez, and J. Tapiador, "Trouble Over-The-Air: An Analysis of FOTA Apps in the Android Ecosystem," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 1606–1622, https://doi.org/10.1109/SP40001.2021.00095.

[25] P. Laperdrix, N. Mehanna, A. Durey, and W. Rudametkin, "The price to play: A privacy analysis of free and paid games in the android ecosystem," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3440–3449.

[26] R. Kumar, A. Virkud, R. S. Raman, A. Prakash, and R. Ensafi, "A Large-scale Investigation into Geodifferences in Mobile Apps," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, 2022, pp. 1203–1220.

[27] Y. Shen, P.-A. Vervier, and G. Stringhini, "A Large-scale Temporal Measurement of Android Malicious Apps: Persistence, Migration, and Lessons Learned," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1167–1184.

[28] K. Khanmohammadi, N. Ebrahimi, A. Hamou-Lhadj, and R. Khoury, "Empirical study of android repackaged applications," *Empirical Software Engineering*, vol. 24, pp. 3587–3629, 2019, https://doi.org/10.1007/s10664-019-09760-3.

[29] A. Aswini and P. Vinod, "Droid permission miner: Mining prominent permissions for android malware analysis," in *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*. IEEE, 2014, pp. 81–86, https://doi.org/10.1109/ICADIWT.2014.6814679.

[30] K. Sokolova, C. Perez, and M. Lemercier, "Android application classification and anomaly detection with graph-based permission patterns," *Decision Support Systems*, vol. 93, pp. 62–76, 2017, https://doi.org/10.1016/j.dss.2016.09.006.

[31] A. A. Taha and S. J. Malebary, "Hybrid Classification of Android Malware Based on Fuzzy Clustering and the Gradient Boosting Machine," *Neural Computing and Applications*, vol. 33, no. 12, pp. 6721–6732, 2021, https://doi.org/10.1007/s00521-020-05450-0.

[32] F. Alswaina and K. Elleithy, "Android malware permission-based multi-class classification using extremely randomized trees," *IEEE Access*, vol. 6, pp. 76 217–76 227, 2018.

[33] X. Li, J. Liu, Y. Huo, R. Zhang, and Y. Yao, "An Android malware detection method based on AndroidManifest file," in *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*. IEEE, 2016, pp. 239–243.

[34] G. Suarez-Tangil, S. K. Dash, M. Ahmadi, J. Kinder, G. Giacinto, and L. Cavallaro, "DroidSieve: Fast and Accurate Classification of Obfuscated Android Malware," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, 2017, pp. 309–320, https://doi.org/10.1145/3029806.3029825.

[35] Y. Aafer, W. Du, and H. Yin, "Droidapiminer: Mining api-level features for robust malware detection in android," in *International conference on security and privacy in communication systems*. Springer, 2013, pp. 86–103.

[36] S. Zhu, Z. Zhang, L. Yang, L. Song, and G. Wang, "Benchmarking label dynamics of VirusTotal engines," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 2081–2083.

[37] W. J. Buchanan, S. Chiale, and R. Macfarlane, "A methodology for the security evaluation within third-party android marketplaces," *Digital Investigation*, vol. 23, pp. 88–98, 2017.

[38] P. Peng, L. Yang, L. Song, and G. Wang, "Opening the Blackbox of VirusTotal: Analyzing Online Phishing Scan Engines," in *Proceedings of the Internet Measurement Conference*, ser. IMC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 478–485, https://doi.org/10.1145/3355369.3355585.

[39] A. Kantchelian, M. C. Tschantz, S. Afroz, B. Miller, V. Shankar, R. Bachwani, A. D. Joseph, and J. D. Tygar, "Better malware ground truth: Techniques for weighting anti-virus vendor labels," in *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, 2015, pp. 45–56.

[40] A. Salem, "Towards accurate labeling of Android apps for reliable malware detection," in *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 269–280. [Online]. Available: https://doi.org/10.1145/3422337.3447849

[41] Q. Wang, J. Li, Y. Zhang, H. Wang, Y. Hu, B. Li, and D. Gu, "NativeSpeaker: Identifying Crypto Misuses in Android Native Code Libraries," in *International Conference on Information Security and Cryptology*. Springer, 2018, pp. 301–320.

[42] K. Chen, X. Wang, Y. Chen, P. Wang, Y. Lee, X. Wang, B. Ma, A. Wang, Y. Zhang, and W. Zou, "Following Devil's Footprints: Cross-Platform Analysis of Potentially Harmful Libraries on Android and iOS," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 357–376, https://doi.org/10.1109/SP.2016.29.

[43] C. Goodwin and S. Woolley, "Sideloading: An Exploration of Drivers and Motivations," in *35th International BCS Human-Computer Interaction Conference 35*, 2022, pp. 1–6, http://doi.org/10.14236/ewic/HCI2022.37.