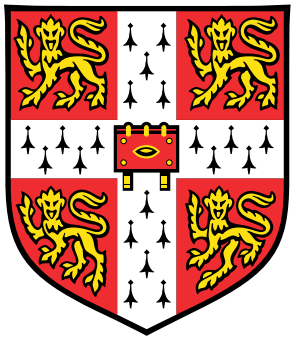
The effects of unauthorised software modifications on smartphone users and developers



Luis Adán Saavedra del Toro

Department of Computer Science and Technology
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Queens' College

September 2025

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the preface and specified in the text. It is not substantially the same as any work that has already been submitted, or is being concurrently submitted, for any degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Luis Adán Saavedra del Toro
September 2025

Abstract

The effects of unauthorised software modifications on smartphone users and developers
Luis Adán Saavedra del Toro

The market leading smartphone operating systems, Android and iOS, allow users to install apps through official pre-installed markets. Android also supports app installation from third-party sources, fostering competition and enabling open source app markets. However, this also enables the proliferation of markets distributing pirated and modded apps: apps whose features and functionality have been altered by a third-party. Modded apps typically claim to offer users premium or subscription features for free, no ads, free in-app purchases, additional in-game resources, etc.

We present the results of the first large-scale study of Android modded app markets covering over 146 000 modded apps from the 13 most popular modded app markets. Despite the common belief that sideloading requires a jailbroken iPhone, we show this is not the case and present the first study of the iOS modded app ecosystem, covering over 40 000 apps from the 9 most popular modded markets.

Original app developers lose significant potential revenue from modded apps due to the free provision of paid apps; the free availability of premium features that require payment in the official app; and changes to advertising identifiers, which took place in 21% of the Android apps with ad IDs. While users benefit from increased competition and free pirated and modded apps, these apps pose risks to user privacy and security. Modded apps are significantly riskier than their official versions: modded Android and iOS apps are 10 and 33 times more likely to be malicious, respectively.

We survey modded app market operators and 717 app developers affected by modded apps. Modded market operators have economic incentives to break copyright law and make it difficult to file complaints. They perform little to no security testing of the apps they host and benefit from app developers' intellectual property. Meanwhile, original developers suffer losses from missed purchases, reduced advertising revenue, additional support requests, and reputational damage. Unfortunately, developers find legal protections are ineffective at preventing modded versions of their apps appearing on third-party stores. Developers are unaware of, or find it hard to use the security features and technical tools which can make the production and use of modded apps much harder.

We conclude with a review of the technical and legal methods hardware and OS vendors, developers and regulators can use to tackle modded apps with the aim of better protecting developers' intellectual property and revenue as well as user security and privacy.

Acknowledgements

First, I must thank my supervisor, Alastair Beresford. I am forever grateful to him for his incredibly helpful guidance and support during my MPhil and PhD. I could not have asked for a supervisor who provided a better balance of clear direction when needed and the freedom to explore independently, both of which have made me the researcher that I am today.

I would also like to acknowledge and thank my collaborators and lab mates who made my PhD more enjoyable and inspiring. Alice, Hridoy, Alex, Ceren, Diana, Jenny, Laurie, Martin, Michael, Nicholas, Richard, Stan, and everyone in the Security group.

I owe everything to my family, especially my parents, Carmen and Luis. Their unwavering support and encouragement have made all the difference and inspired and allowed me to pursue my goals. Thank you to all my family and friends back home.

Many people have made my time at Cambridge even more memorable, mainly my partner and friends. Thank you Anisha, Alberto, Alicia, Carlos, Celia, Charlie, Cindy, Clara, Cristian, Diego, Elliot, Jorge, Kai, Lambert, Nick, Ryan, and many others.

Thanks also to Fundación MAPFRE Guanarteme for funding my MPhil, Nokia Bell Labs for generously funding my PhD and Google ASPIRE for granting our project extra funding.

Finally, thank you, dear reader, for taking the time to read this dissertation.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Smartphones are ubiquitous nowadays and offer users and developers endless possibilities thanks to apps, one of the most popular types of software. They give users access to their bank accounts, social media, secure messaging, work and personal files, health monitoring, etc. Apps are typically distributed and installed through the official pre-installed app markets: the App Store in iOS and Google Play in Android. Many Android smartphone manufacturers include their own app store alongside Google Play, such as Samsung's Galaxy App Store, Huawei AppGallery, Xiaomi GetApps, etc. These markets provide users quality assurance through user reviews, app vetting and swift removals of dangerous apps, etc. They offer developers tools for easier development, billing, and instant access to millions of customers.

Furthermore, Android has an open design philosophy. This includes allowing users to sideload apps, i.e. to obtain and install application packages (APK files in Android) from third-party markets and websites or to install apps shared as (APK) files between users. This has enabled the creation of markets focused on open-source software like F-Droid, and independent markets like Aptoide. However, it can also open up users to scams and malicious apps, and has led to the existence of third-party markets filled with pirated apps and apps with modified code. *Modded apps* are unofficial versions of apps whose code, metadata, features and/or functionality have been altered by an unauthorised developer or third-party. Modifications may include: free subscription features, unlocked in-app purchases (IAPs), in-app or in-game perks (virtual money, extra lifes, etc.), no ads, or access to paid apps without charge.

Apple's mobile operating system, iOS, does not allow users to sideload apps by default, unlike Android. However, despite its 'walled garden' approach and the common belief that a jailbroken (rooted) iPhone is needed to sideload apps, sideloading and modded apps are a reality in iOS. Sideloading has been possible in consumer laptop and desktop operating systems for a long time. It gives users and developers more choice, and allows developers to sign their own apps and sell them, extra features and subscriptions without paying a percentage of revenue to monopolist markets. However, as we will show, most of the markets distributing modded apps do not vet apps to the same standard as Google's or Apple's official app markets.

This dissertation aims to give the first thorough overview of the modded app ecosystem, by approaching it from different perspectives: the ecosystem of Android modded app markets and the apps hosted in them (Chapter 3); the iOS sideloading and modded app markets and the apps they offer (Chapter 4); the modded market operators' motivations, and the original app developers affected by modded apps and their awareness, revenue losses, impacts suffered and experience with available techniques to prevent modding and piracy (Chapter 5); and finally the current takedown notices (DMCA) process available to copyright holders and its shortcomings, and possible technical and legal solutions to tackle modding (Chapter 5).

This work is very timely because, while there is prior research on manufacturer-specific and third-party markets in Android, there is no research on modded app markets and how they affect user security and the original developers. Similarly, there is very little research on sideloading in iOS and nothing we are aware of on the operation of modded iOS app markets, the risks introduced by the installation of the apps hosted in them for the users, and the effects on the original app developers' revenue streams and reputation. Furthermore, Apple has faced increasing pressure from both the EU and US legal systems over its closed ecosystem and monopolistic policies. The EU designated Apple as a 'gatekeeper' under their Digital Markets Act (DMA) in 2023 [32, 79]. Among other requirements, Apple must open their ecosystem to third-party app markets in Europe [27, 78, 80]. This dissertation is relevant for regulators, who need to balance competition and fair markets with user security and privacy and developers' intellectual property protection. Our results and recommendations are also important for Apple, Google and app developers wanting to protect users from malicious modded apps. They are of course also useful for users themselves who want to be informed of the risks and effects these apps can have on their privacy and original developers' revenue.

In this dissertation, I introduce *ModZoo*, a dataset of over 146 000 modded apps collected over 3 months from the 13 most popular Android modded app markets (out of over 400) and the first large-scale study of Android modded apps. This dissertation also introduces *iOSModZoo*, a dataset of over 40 000 modded apps collected from the 9 most popular iOS modded markets (out of 89) over 11 months and *iOSZoo*, a dataset of over 77 000 related iOS App Store apps.

We surveyed 19 modded market operators and 717 original app developers. These 717 survey respondents are Android developers of all sizes whose apps we found in modded app markets. Half of the 717 respondents are also iOS developers. Most were aware of Android and even iOS modded apps. Our results shed light on how each party involved in mobile app distribution is affected.

Original app developers technically benefit from sideloading as it allows them to choose alternative distribution methods and have different versions of their apps, benefitting from market competition to get better conditions and pay smaller commissions. However, our technical modded app analysis shows they lose significant potential revenue due to the provision of paid apps for free: around 5% of apps on Android and almost 30% on iOS modded markets are pirated paid apps and games offered for free. We found that pirated paid iOS apps and games had over 1 billion downloads in just three iOS markets. Furthermore, many modded apps offer subscription features and in-app items and purchases for free. Android modded apps also have modified advertising identifiers (in

21% of the apps with ad IDs), thus redirecting ad revenue away from the original developers to third parties.

Original app developers who responded to our survey reported revenue losses from IAPs, advertising, sales, increased server calls, and increased third-party API use within their apps. They also reported reputational damage, increased errors and crashes, and additional support requests. Mods even demoralised some developers. Less than half of the developers who responded use techniques to combat modding and unauthorised redistribution, mainly due to lacking technical expertise, finding that they do not work or they cause problems for legitimate users.

Users obviously benefit from increased competition and choice, the availability of pirated and modded apps for free, and better compatibility with older devices. However, user security and privacy are also affected by modded apps. Our findings show the installation of modded Android and iOS apps is significantly riskier than official versions. Android apps with modified code are 10 times more likely to be marked as malicious by VirusTotal than their Google Play counterparts, and they often request additional permissions including dangerous and deprecated ones. Modded iOS apps are 33 times more likely to be classified as malicious than their App Store counterparts. Furthermore, installation methods used by most modded app markets directly install apps without giving users access to the files before or after installation. A popular distribution method technically allows the market operator to remotely extract private data from and install apps on users' devices without user interaction.

Modded apps also negatively affect **legitimate users** of the original versions: multiplayer game players experience unfair disadvantages when compared to modded users with access to game cheats, and some modded messaging and social media apps allow users to record, screenshot, or save content without notifying the user that sent or posted it. Some mods allow users to see others' connected status without showing theirs, see deleted messages, and save 'disappearing' or 'view-only once' pictures, etc. Anti-modding measures used by developers can also disrupt legitimate users' experience, such as requiring an internet connection at all times or reducing an app's compatibility with older devices.

Third-party markets hosting modded apps benefit their operators. A unique catalogue of pirated and modded apps can attract more visits and generate them advertising revenue. Many iOS modded markets even offer paid subscriptions for better mods and fewer ads. We find modded market operators that responded to our survey are primarily motivated by monetary gains, mainly through ads on their markets. Most of the 19 operators self-declare they perform no security testing beyond scanning apps with third-party tools before adding them to the market, and only a few claimed they performed periodic scanning of their catalogue. This leads to security concerns for **developers and users**; users are enticed by modded features found in these markets to install many outdated, untested, and illegally modified versions of popular apps. We find many Android and iOS modded markets operate as forums where admins and users can upload apps as files for others to download, making it easier to keep an updated modded app catalogue but very difficult for these markets to check the security of all apps they host. Worryingly, our findings point to a growing number of modded apps

and modders in both Android and iOS markets. User engagement data reveal a steady active user base in two iOS markets over the last four years.

Official app market operators and OS vendors benefit very little from sideloading, however it can gain them favour with users and regulators. It can erode developer confidence in the market and OS. Our survey shows many developers view Apple more favourably than Google because of their handling of sideloading. Official app markets can also see their commission revenue affected by fewer purchases and developers choosing sideloading markets to publish their apps. Smartphone manufacturers are typically also the official app market operator or include one or more official markets installed by default.

Based on these results, in this dissertation I propose Apple and Google should increase awareness of the existing tools they have already made available to developers [8, 21, 24, 33, 45]. Google and Android manufacturers should guarantee extended device support to ensure wider availability of the latest tools and APIs beyond the first two years [48]. Apple should allow client-side app scanning, similar to Google Play Protect [100, 104], now that third-party markets and sideloading are more prevalent in iOS. Additionally, ad revenue provides the majority of funding for modded markets (market website ads), modders (in-app ads), and even Google (search, market website, and in-app). This creates a complex ecosystem where ad networks do little to prevent the use of ads on modded apps and markets.

The DMCA process is clearly not fit for purpose as it works today: only 2% of the developers we surveyed were successful with DMCA takedown notices. Most markets knowingly host content that infringes copyright and the market operators we surveyed host that content unafraid of any consequences. The vague, menacing wording in most markets suggests they use DMCA mainly as a show of compliance and even a deterrent for developers. Developers, especially smaller ones, cannot keep track of the hundreds of third-party markets hosting multiple modded versions of their apps. Legislative changes may be needed to more effectively and lastingly remove modded apps and modded app markets from search results and to prevent ads from appearing on them.

1.1 Publications

Parts of this thesis are based on research projects done in collaboration with others, and published or under submission in different papers:

- Luis A. Saavedra, Hridoy S. Dutta, Alastair R. Beresford and Alice Hutchings. **ModZoo: A Large-Scale Study of Modded Android Apps and their Markets** [176]. In Proceedings of the 2024 APWG Symposium on Electronic Crime Research (eCrime), pp. 162–174, 2024.

  The *ModZoo* paper forms the basis of Chapter 3. Hridoy prepared the list of markets and a prototype scraper, we expanded the list and did the case study of apps together. I ranked the markets and created the first scraper of each type. Hridoy helped adapt them to other markets, oversaw the apps and metadata scraping and did the metadata analysis. I parallelised and automated the scraping further, designed the databases, ran the app analysis, summarised the findings, and was the primary author of the paper. Alice and Alastair contributed towards the development of the ideas and their presentation.

- Luis A. Saavedra, Hridoy S. Dutta, Alastair R. Beresford and Alice Hutchings. **iOS[Mod]Zoo: A Large-Scale Study of Third-Party iOS App Markets**. Under submission.

  The *iOSModZoo* paper is the basis for Chapter 4. Hridoy and I used the same methodology as in the Android study to find around 90 iOS modded app markets. I ranked them and prepared an App Store scraper and a scraper for the 9 most popular markets, as well as the scraping pipeline and databases. Hridoy ran the scrapers and adapted them as markets changed. He scraped the App Store metadata and condensed the features of interest I had identified. I ran the app analysis, case study and summarised our findings, including the metadata analysis findings. I was the primary author of the text. Alice and Alastair contributed towards the development of the ideas and their presentation.

- Luis A. Saavedra, Hridoy S. Dutta, Alastair R. Beresford and Alice Hutchings. **App-solutely Modded: Surveying Modded App Market Operators and Original App Developers** [177]. In Proceedings of the 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS '25), pp. 739–758, 2025.

  The paper forms the basis of Chapter 5. Hridoy ran the modded market operators surveys and I summarised all the responses. Hridoy and I obtained the contact details from Google Play of all app developers whose apps were in ModZoo. I wrote the questionnaire and contact emails and finalised them based on feedback from all co-authors. I ran the survey and coded the thousands of email responses and 717 survey responses. Alice used my codebook to code again a 10% random sample for robustness. I studied modded markets' DMCA pages and the takedown requests sent by and mentioning each of the markets, and was the primary author. Alice and Alastair contributed towards the development of the ideas and their presentation.

1.2 Contributions

This dissertation makes the following contributions:

- In Chapter 3, I present an overview of the modded app ecosystem and the results of the first large-scale study into the 13 most popular Android markets that offer *modded* apps. I collected 146 thousand modded Android apps (*ModZoo* dataset) and matched them with their Google Play counterparts. Around 90% of apps are modified in some form and 75% have modified code. Modded app markets likely cause reductions in the original app developers' income due to widespread availability of paid apps and premium features for free, and redirection of ad revenue, including 21% of apps with altered ad IDs. Furthermore, modded Android apps are riskier for consumers: 23% of modded apps request additional permissions and apps with modded code are 10 times more likely to be marked as malicious by VirusTotal.

- In Chapter 4, I present an overview of the iOS sideloading and modded app ecosystem and the first in-depth study into the 9 most popular iOS modded markets. I collected 40 389 modded iOS apps over an 11-month period (*iOSModZoo* dataset), and 77 189 App Store apps (*iOSZoo* dataset). Modded apps likely reduce income for app developers and official markets. Many modded markets have premium subscriptions. Almost 30% of the modded apps are pirated paid apps offered for free, they have a total of 1.12 billion downloads. We found 221 Apple Arcade games (from a catalogue of 'over 200'). Modded iOS apps are much riskier for consumers: they are 33 times more likely to be classified as malicious by VirusTotal than App Store apps; there are no tools for users to analyse iOS apps before installation; markets sign 78% of the apps with enterprise certificates, others use device management (MDM) profiles which can extract private data and remotely install apps in users' iPhones.

- In Chapter 5, I present the results of the first survey-based study of Android and iOS modded market operators (§5.2): they have economic incentives, mainly monetise their markets through ads, perform little security testing of apps and have misconceptions regarding iOS and Android security. Second, I present the results of the first survey-based study of 717 original app developers affected by modding (§5.3): developers are very familiar with Android modded apps, but less so with iOS modded apps. Developers of all sizes suffer revenue losses; increased costs and support requests; and reputation damage. Third, I present a study of DMCA compliance of the 23 modded markets studied in Chapters 3 and 4, confirming the developers' survey findings that DMCA is not usable (§5.4). Finally, I propose solutions to the challenges presented by modded apps markets for developers, manufacturers, and legislators (§5.5).

1.3 Research questions

The overarching research questions (RQ1–RQ4) this thesis aims to answer are the following. Each chapter also addresses subquestions that relate back to these main research questions. Questions prefixed with 'RQa' concern Android (and are answered in Chapter 3), those with 'RQi' concern iOS (Chapter 4), and 'RQs' relate to the market operator and original developers surveys (Chapter 5). The first three research questions are expanded into more detailed subquestions and partially answered in the Android and iOS chapters. The surveys complement the previous chapters by addressing RQ1–RQ3 in a supplementary manner as well as answering RQ4.

RQ1 What do the modded markets and apps ecosystems look like? How do they differ between Android and iOS?

- RQa1 What is the ecosystem of modded Android markets and apps composed of, and what is its size?
- RQi1 What is the ecosystem of modded iOS markets and apps composed of, and what is its size?
- RQi4 How does the ecosystem of modded iOS apps differ from its Android counterpart?
- RQs1 What motivates operators to run modded app markets?

RQ2 What are the financial incentives for operating modded app markets?

- RQa2 What are the financial incentives for operating modded Android app markets? How does this affect the original developers and markets?
- RQi2 What are the financial incentives for operating modded iOS app markets? How does this affect the original developers and markets?
- RQs2 What are modded market operators' revenue streams?

RQ3 How are users, original developers, and app markets affected by modded apps?

- RQa3 What are the security implications of installing apps from these Android markets?
- RQi3 What are the security implications of installing apps from these iOS markets?
- RQs3 What types of security testing do they perform, if any?
- RQs4 Which developers have had their apps modded?
- RQs5 Are developers aware of modded app markets?
- RQs6 How do modded and pirated versions affect developers?

RQ4 What technical and legal tools are available to developers and are they effective? What are the possible solutions to unauthorised software modifications?

- RQs7 What tools are available and used by developers?
- RQs8 Are DMCA notices effective and used by developers?

1.4 Ethical considerations

The research described in this dissertation has been conducted following the University of Cambridge's ethical research policy. Where appropriate, approval was sought from the Ethics Committee of the University of Cambridge Department of Computer Science and Technology prior to undertaking the research work (CST Ethics Committee reviews No. 1952 and 2276). These ethics reviews covered the surveys in Chapter 5: the survey instruments, data handling, contact procedure, etc. The wider project and app scraping plans were described to give more context on the overarching projects, but the main focus of the ethics reviews were the human subject studies.

We followed existing procedures to handle our app datasets. Our datasets only include publicly-available apps and metadata. The datasets resulting from our research are only available to bona fide researchers through the Cambridge Cybercrime Centre under a licence agreement between our institutions.[1] The last thing we want is to create another modded app market! This licence agreement ensures that only legitimate researchers with a research interest have access to the Cybercrime Centre's datasets, that they handle the data properly, and that no results are published without being checked by the centre for potential breaches.

The surveys presented in this dissertation were all run on voluntary participants, invited through their publicly-available contact details.

More detailed ethical considerations and practices are included in each relevant chapter.

[1] https://www.cambridgecybercrime.uk/process.html

Chapter 2

Background

This chapter gives an overview of the history of software distribution and the different aspects and challenges that have led to the current mobile app distribution and security landscape. We present an overview of software distribution over time, starting from physical media to mobile app stores and how the roles of users, developers, and manufacturers shifted over time in §2.1. We then give an overview of the security mechanisms in Google and Apple's official app stores in §2.2, and the existing sideloading techniques in iOS in §2.3. There are many different ways of sideloading in iOS, despite the popular belief that jailbreaking is the only way of sideloading apps. There are some caveats not unlike Android, as we will discuss. Finally, we dive into the European Union's and more recently the United States' legal battles with Apple (and to a lesser extent Google) and the questions it opens about consumer choice versus control and security in §2.4. By the end of this dissertation we hope to have answered some of these questions.

2.1 Brief history of software distribution

In the early days of computers, users were also the developers and machine owners (and in the 1950s and early 1960s sometimes also the machine designers and builders). What we now know as software and its distribution evolved from physical media such as punch cards and punched tapes to magnetic tapes, then floppy disks, CD-ROMs, DVDs and other physical media gave way to digital distribution [62, 65]. Subscription services also became very popular in the 2000s overtaking physical copies with lifetime licences [63].

Eventually, computers (mobile devices) became small enough to carry in our pockets, and mobile apps and games have become a lucrative product for developers. These are even more lucrative for the companies that control the markets: Google and Apple (and a few smaller ones).

2.1.1 Physical media

In the early days of computing, users tended to be the machine owners, (designers and builders sometimes) so they wrote their computer programmes themselves. Otherwise, computer programmes or what we now know as 'software' came bundled with computers or custom developed for each customer [64, 70, 111] and was loaded using punch cards or punched paper tapes [56, 105]. Thus, software was 'free' when buying the hardware, and consumers/users trusted its provenance implicitly.

Punched cards had to be discarded if an error was made or once they were not needed, as they were not reusable. They did not have a standard format to represent data until they were standardised in 1969. Shortly after that they were completely replaced by magnetic tapes and computers [105].

Magnetic tapes and minicomputers (1960s–1970s)

Minicomputers were a range of medium-sized computers that filled the gap between big and expensive mainframe computers and smaller and less powerful personal computers. Minicomputers allowed more businesses to access more affordable computing, leading to more software being needed and distributed [89, 139]. By the 1960s IBM had amassed a vast, free software library, some coming from their users who were willing to share programmes. Following an antitrust lawsuit against IBM in the US, in 1969 IBM unbundled their hardware and software. There were already other smaller independent companies that offered software products to mainframe computer users. However, IBMs unbundling turned software from a free by-product included with every machine into a competitive business of its own [64].

Magnetic tapes and early disk drives were used for software distribution, the concept of software licencing began when vendors offered software and hardware separately [89, 139]. However, authentication was still based on physical markers such as vendor labels, manuals and branding, and direct or in-store distribution. Magnetic tapes have remained in use in data centres and to store large amounts of cold data (large amounts of data that are rarely accessed, e.g. large redundant backups) due to their lower costs compared to even hard disks, their low energy consumption, and increased security against cyberattacks, despite their slower speed [133].

Floppy disks and microcomputers (1970s–1980s)

Magnetic tapes evolved into floppy disks in the 1970s, with IBM introducing the 8-inch floppy disk, followed by smaller formats that became popular with personal computer users: first a 5.25-inch format, then a 3.5-inch format by Sony in the 1980s. Magnetic tapes slowly got phased out in personal computers, with notable exceptions including the BBC Micro computer. Designed to be an accessible and affordable home computer and introduced in 1981, the BBC Micro computer used cassette tapes to store programmes input by the users instead of the more expensive floppy disks [39]. It was also possible for users to copy code by hand from specialist magazines that included code to be copied straight into the users' machines [175].

The introduction of popular personal computers (PCs) such as the Altair 8800,[1] the IBM PC,[2] and the Apple II,[3] made floppy disks ubiquitous [4]. Floppy disks became the most common distribution medium for software. They also made copying and sharing easier among consumers, with some hobbyist software being distributed as type-in code sent as booklets. A prominent example is Microchess, a game from 1976, initially distributed via mail as source-code in a manual booklet and later available also on tape [117].

However, using floppy disks to distribute software quickly led to software piracy, since users could easily create copies of the software programmes they had purchased. There were some protections such as physical dongles, licence agreements or product keys, however, these were typically easy to circumvent for user-oriented software. Games had more inventive methods during the 1980s and 1990s of checking the user had access to the original manual or packaging. These included requiring a specific word from the manual chosen at random, including vital hints required to progress the game in the manual, and similar interactive checks [122]. Another type of protection included deliberate bad sectors or weak bits that caused standard copying tools to determine there was an error in the disk and terminate the copy [82]. This led to systems such as ddrescue[4] and techniques to copy disks analogously.

CD-ROMs and DVDs (1980s–2000s)

The next leap came with CDs, developed by Sony and Philips in 1982, and with the first CD-ROM software product, Grolier's Electronic Encyclopedia, released in 1985 [62]. Again, licences and product keys were the standard, and the physical packaging and retail methods assured users of the software's provenance [69]. With slow internet speeds in the 1990s, the more common licencing models were *buy once, own forever* and trialware, i.e. limited demos. Subscription models remained uncommon. However, similar to floppy disks, CDs posed piracy problems, so anti-piracy methods were used for CDs similar to those described above for floppy disks containing games. Other methods included the first uses of online activation used by Microsoft for their OSs at the end of the 1990s [121]. However, many of these early DRM protections were quickly cracked or very intrusive to users, as detailed below.

2.1.2 Digital online distribution and DRM

As early as the 1980s hobbyists exchanged shareware on Bulletin Board Systems. However, some of these also hosted pirated content.[5] Digital distribution became popular in the late 1990s and early 2000s, led mainly by the music industry. The MP3 format meant decent-quality music files were now easy to send without requiring substantial storage capacity [5]. Something similar would happen with

[1] https://archive.org/details/historyofmodernc00ceru_0/page/226/
[2] https://archive.org/details/historyofmodernc00ceru_0/page/268/
[3] https://archive.org/details/historyofmodernc00ceru_0/page/264/
[4] https://www.gnu.org/software/ddrescue/
[5] http://textfiles.com/bbs/r&estory.txt

DVDs. There were also peer-to-peer file sharing networks such as Napster, launched in 1999, which was obviously used for piracy and thus shut down in the early 2000s.

However, with the ease of access and copying of CDs and DVDs companies sought to protect the artists' copyright and their revenue by including digital rights management software (DRM). Microsoft used DRM in Windows Media Player since the 1990s with Windows 98, replacing it with 'PlayReady', a newer version launched with Windows 10. Apple's 'FairPlay' was launched with the iPod and Apple's QuickTime media player. Similar to Microsoft's DRM solution, it encrypted songs under master keys [5]. There have been controversies with some of these DRM software solutions over the years, the most notable was a Sony DRM solution 'XCP' included with all their CDs that was found in 2005. Sony's tool ejected the disk if the user did not agree to the terms, otherwise it installed a rootkit unbeknown to users, spied on them, and even introduced vulnerabilities to Windows, on top of that it also could not be uninstalled [180, 181]. The backlash against Sony was accompanied by lawsuits that compelled Sony to completely retract its DRM measures after reaching a settlement [75]. Furthermore, DRM in music had not prevented piracy, and DRM was unpopular with users who wanted full ownership of the music they purchased. In 2007 EMI and Apple were the last to abolish DRM in downloaded music, putting an end to DRM in CDs and music distributed online [29, 151]. By then, some DRMs in DVDs were broken within days and the development costs and burden on the users were not justified by sales or copy prevention [71].

HTML5 supports multimedia content with DRM and communication with online licence managers, this is used by Netflix and others to make it more difficult to copy their content. Apple, Amazon, Netflix and others still use DRM to make it difficult for users to copy songs, apps, ebooks, and movies [5]. Recently, Signal used a DRM flag in Windows to prevent Windows' Recall feature from taking periodic screenshots of private conversations [140].

2.1.3 Gaming and DRM

Games were one of the first applications of computers, with the EDSAC, the world's first computer, captivating the imagination of research students at the time who immediately wrote games for it [5]. The main way gaming companies tackled safe distribution and illegal copies of games since the 1980s was through hardware protection. The market was dominated by the likes of Sega, Nintendo, Atari, etc. (and later PlayStation and Xbox in the late 90s and early 2000s) that had consoles with closed architectures and full control of the proprietary cartridges. Thus, gaming consoles were typically sold at a loss, with the sales of software cartridges and later discs containing games and other add-ons subsiding the costs and generating revenue through that generation of consoles' existence. Examples of these are Nintendo's home consoles and Microsoft's Xbox and Sony's PlayStation consoles [5]. Many of them eventually got hacked by users wanting to run other software in them including emulators of older consoles, and this also led to piracy [41, 57, 66, 222].

Thus, gaming used physical protections as long as possible (gaming consoles still do), and then moved on to DRM and other techniques when digital distribution and PC gaming became one of

the main gaming markets. The PC games marketplace Steam was one of the first to push for stricter purchaser authentication beyond simple CD key-based authentication to combat people easily burning CDs at home in 2003 [51, 148]. PC games typically use internet connectivity to periodically check for the correct licence, others go beyond that by moving some of the functionality to servers and requiring an internet connection to play. This makes games more difficult to crack but there are also infamous examples like Denuvo which is generally assumed to hinder performance and hated by consumers [142, 154]. There is less recent evidence of performance impacts [126, 147].

For these and more reasons like some DRM making games require constant connection to the internet to work, defunct companies making older games unplayable, problems with updated hardware [205], etc. other providers have focused their stores around non-DRM games only [132] and some developers remove DRM months or years after launch [127].

2.1.4 History of mobile phones

The first 'mobile' phones were large units installed in cars and heavy briefcases from the late 1950s [173]. Then came the Motorola brick, first commercialised in 1984 and weighing over 1 kg, it was the first truly *portable* mobile phone. It had a battery that provided 30 min of talk time and took 10 h to fully charge. However, these used unencrypted analogue communications to the cell towers, which made them vulnerable to passive eavesdropping and impersonation attacks [5].

The GSM standard was developed in the late 1980s and released in the early 1990s. The GSM standard (2G) adopted digital communication and used SIM cards to authenticate user devices and encrypt communications between the handheld device and the base stations [5]. There were some security issues arising from the base stations (cell towers) not being authenticated, thus impersonating them and intercepting calls was possible for law enforcement or motivated adversaries [5]. With better battery technology and better cell coverage mobile devices could get smaller while offering better reception and battery life. Soon appeared the Motorola flip phones, Nokia 3310s, and new mobile phone companies. These phones came with preinstalled software allowing calls, SMS texting (thanks to 2G GSM), calendar, etc. Some would include games such as the popular Snake, however all of these phones had a limited set of features and apps which were restricted and known at the time of purchase. For a while ringtones where the only available downloadable purchases for mobile devices.

Nokia and other manufacturers including Sony Ericsson, Samsung, and Motorola used Symbian OS in the 2000s. Nokia purchased it in 2008 and even added a game store in their version of it. However, games and apps in general never took off properly in Symbian OS due to how difficult it was to develop for it [158, 168]. Soon came Apple's iOS (originally named iPhone OS) and later Google's Android with app markets, better developer tools and less compatibility issues, ensuring apps worked in multiple devices seamlessly (in the case of Android). Microsoft's Windows Phone to a lesser extent also appeared, eventually even Nokia stopped using Symbian OS and moved their phones to use Windows Phone after partnering with Microsoft in 2011 [150]. Nokia's devices and services unit was later bought by Microsoft in 2014 [164]. These app markets also offer users peace

of mind in terms of security, since all apps in the App Store and Google Play must pass manual and automated reviews.

The introduction of app markets in iOS was one of the first times in the history of computing that a closed-market was seen as an advantageous feature, and sideloading or installing user-written programmes was discouraged. Google's Android operating system allowed sideloading alongside their official app market. However, they announced upcoming limitations in August 2025 for certified Android devices, disallowing sideloading of apps written by unverified developers [88], before clarifying in November 2025 that "experienced users" will be able to go through a new series of warnings and information screens before accepting the risks of installing unverified software [86].

Most smartphones quickly adopted 3G communications, which were almost a hundred times faster than 2G, enabling the smartphone revolution [5]. It also had more robust security, and was succeeded by 4G and 5G which were even faster and enabled even more powerful apps to work seamlessly.

2.2 Security in Google Play and the App Store

Google and Apple certify that the apps they host come from the developers associated with them, thus proving to users the apps come from trusted developers, including a contact email in the Google Play and the developer website in the App Store. Developers sign their apps and send them to Google and Apple for testing and distribution. Google and Apple can thus ensure every app hosted in their market comes from a verified developer, before Google and Apple sign the apps again for distribution in their markets. All apps are also tested before being hosted on these markets. It works as follows in each ecosystem.

2.2.1 Signing keys

In Android, developers can manage their own keys and essentially sign their apps before uploading them to Google Play if their app was created before August 2021 [9]. All apps are tested by Google automatically before being hosted in the store, and then retested in customer devices using Google Play Protect after installation.

Developers of all apps created after August 2021 (or those with older apps who want to) have to use 'Google Play Signing', a tool offered by Google where developers can have a local signing key while Google hosts a different key they resign the apps with before publishing them after testing and optimising the APKs.[6] Google stores the keys securely for developers, making them more secure, and allowing key resets. The previous approach in comparison would make a developer unable to update their apps after a key reset.[7]

Apple's approach is more similar to the latter option; all code running on an iPhone must be signed with an Apple-issued certificate. The only exception are businesses wanting to run proprietary apps

[6]https://support.google.com/googleplay/android-developer/answer/9842756
[7]https://developer.android.com/studio/publish/app-signing

on company devices, but even these need to be verified through Apple first [34]. This requirement restricts apps from loading unsigned code resources or using self-modifying code, according to Apple.

Thus, iOS developers must register in the Apple Developer Programme, which verifies their real-world identity before enrolling them and issuing them a certificate. Developers can then use this certificate to sign their apps and submit them to the App Store. Apple then tests the apps before distributing them in the App Store.

2.2.2 App security testing

Both Google and Apple test apps submitted by developers before hosting them in their markets.

Google Play Protect uses machine learning to detect threats, preventing 2.4 million policy-violating apps from being added to Google Play and banning 158 000 malicious developer accounts in 2024 [45, 100]. They also detected 1.3 million apps trying to get "excessive or unnecessary access to sensitive user data" [45]. Besides their cloud-based scanning of every app before it appears on Google Play, Google Play protect provides on-device protection [100, 104]. This real-time on-device scanning tool automatically scans every app downloaded and installed on Android phones periodically regardless of whether they were downloaded from the official Google Play or any third-party market. Thus, it scans over 200 billion apps daily, and in 2024 found over 13 million malicious apps from third-party markets [45]. It also performs periodic scans and updates its information on unwanted programmes over-the-air (OTA), and upon finding potentially harmful apps it notifies the user to remove them, disables them in the meantime, or removes the apps automatically.[8]

Apple on the other hand does not (except in the EU) allow the installation of apps from outside the App Store (sideloading). The only exception to this outside the EU are developer testing tools allowing anyone with a free or paid developer account to test apps on their own devices, businesses can test their apps too. Also, company-owned devices can be used for testing and proprietary apps can be installed by administrators. Facebook famously used this last method to circumvent Apple's App Store policies and distribute their 'Facebook Research' app to users—including teenagers—to collect detailed usage data from their devices in exchange for gift cards [37]. All of these sideloading methods will be discussed in detail in §2.3.

All apps in the App Store come from identified developers (the real-world identity or business is known and documented by Apple), and they all go through automated and human review.[9] Apple states all apps in the App Store have gone through "automated scans for known malware". Followed by a "human review by a team of experts" that review the app description and screenshots to prevent scams. Finally, further manual checks ensure the app is not requesting unnecessary access to sensitive data, and apps targeted at children are checked with extra care against the more strict data collection and safety rules involved [33]. Millions of apps are rejected each year and over $7 billion in potentially

[8] https://support.google.com/googleplay/answer/2812853
[9] https://support.apple.com/guide/security/secf49cad4db

fraudulent transactions was prevented between 2020 and 2023 [30]. Furthermore, apps are sandboxed to prevent them from accessing other apps' data.[10]

Apple admits that while iOS gives users many protections and strong security, "those protections aren't engineered to protect against choices a user might be tricked into making" [33].

2.3 Sideloading methods in iOS

Apple's mobile operating system, iOS does not allow users to sideload apps by default, unlike their main competitor Android which has a more open design philosophy. While Android allows users to easily install apps outside Google Play (sideload), iOS is commonly known as a more closed ecosystem or 'walled garden'.

Recently, Apple has been legally required to open up their OS in the European Union to allow European users to sideload apps and exercise their right to choose a third-party app marketplace from which to download apps without Apple as an intermediary [23, 77, 78]. This and changes to commissions in in-app purchases in the US have come as a result of litigation with third-party companies registered in Europe over Apple's walled-garden approach to app distribution in iOS and the commission fees they charged for each app or in-app purchase done through their App Store [26, 55, 59]. There are questions over how Apple has implemented their sideloading compatibility and the fees they charge third-party markets (see §2.4). Apple was also sued in 2024 by the Justice Department in the United States over their monopolisation of the US smartphone market (see §2.4).[11]

The truth is that sideloading in iOS is now a reality in Europe, but contrary to popular belief and Apple's claims it was already a reality worldwide for years before the EU regulation. This section introduces the sideloading techniques used by these pre-regulation markets. Chapter 4 studies their ecosystem of apps, and the reasons why they will never become part of the regulated third-party markets allowed by Apple and the EU. The sideloading methods discussed below in §2.3.2 are arguably easier and more accessible than jailbreaking both in terms of technical complexity and availability in the latest models of iPhone and iPad, and versions of iOS. These methods are so far unexplored in the literature, unlike jailbreaking.

2.3.1 Jailbreaking is not the only way to sideload

There is a common misconception that jailbreaking is the only way to sideload apps in iOS [80, 81, 92, 144, 202, 203, 212]. However, there were already other more accessible methods of sideloading available before the new EU regulation, although they are mostly less convenient and more technical than their Android counterparts. This section discusses these methods, as well as the new post-regulation sideloading landscape.

[10] https://support.apple.com/guide/security/sec15bfe098e

[11] https://www.justice.gov/archives/opa/pr/justice-department-sues-apple-monopolizing-smartphone-markets

While jailbreaking was a great option for sideloading apps and other 'tweaks' and themes etc. back in the day, it has been unavailable for years for the latest versions of iOS and the latest iPhones. Thus, most users today do not run a jailbreak-compatible version of iOS and do not own an iPhone old enough to be jailbroken [19].

However, the UK Competition and Markets Authority refers to 'jailbreaking' as the only option available to users who want to sideload applications in iOS [203]. Their report concludes that "the [iOS] App Store does not face a competitive constraint from users sideloading apps" and that 'jailbreaking' is the only way of getting around Apple's restriction on sideloading. The EU's Digital Markets Act makes similar claims of unavailability of sideloading in iOS without jailbreaking or tampering with devices [80, 81]. The Netherlands' ACM (Authority for Consumers & Markets) made similar claims [202]. Similar misconceptions are present in previous research [92, 144, 212].

2.3.2 Popular iOS sideloading methods

While getting root access through a jailbreaking tool [199] and downloading mods and apps from Cydia [116] or similar markets is a well-known approach, there are others available that are more commonly used as they work on all recent versions of iOS and recent iPhones, require fewer steps, and seem less intrusive than the available jailbreak tools.

In fact, iOS devices do not need to be 'jailbroken' or tampered with in any way beyond using well-documented and available developer tools provided by Apple. Many of the tools discussed below are recommended by sideloaded apps users and modded app markets themselves. All modded markets misuse the developer tools provided by Apple to enable sideloading with different drawbacks and against the terms and conditions of Apple's developer programme.

Free and paid developer certificates

Any free Apple ID account can be used as a free Apple Developer account, thus, all iOS, iPadOS, and MacOS users already have such an account by default. This free Apple Developer account is meant to be used by first-time or not established developers, students, etc. to locally test apps they have written (in their laptops or desktop Apple computers) on their iPhones or iPads. This allows apps to be signed by the user's Mac for testing on their own devices. However, free Apple Developer accounts (i.e. regular Apple IDs) have a self-signed app expiration period of 7 days. In contrast, paid developer accounts are also available for $99/year to any Apple user.[12] These paid accounts also allow apps to be self-signed by the user (developer) for testing on their own devices with a longer expiration period of 365 days. Paid developer accounts can be used to install an unlimited number of sideloaded apps at a time on a device compared to just 10 for free accounts, and can install them via cable or over-the-air (OTA), compared to just via cable for free accounts [197]. However, it should be noted that these 10 include e.g. for Spotify, its notification service, 'Now Playing' widget, and 2 other extensions, totalling 5 app IDs, depending on the way the app is implemented and the capabilities it needs.

[12] https://developer.apple.com/support/compare-memberships/

AltStore and developer certificates

A commonly-used tool recommended by many users and also by many of the markets is 'AltStore'.[13] It was created by Riley Testut as a way to distribute his Game Boy Advanced iOS emulator (GBA4iOS) and later 'Delta' to non-jailbroken devices. Initially, Apple allegedly told him his emulator would be allowed in the App Store, but after it was developed, Apple did not allow it to be distributed through the App Store as it had introduced a policy against emulators.

AltStore takes advantage of the developer tools and testing process to sign sideloaded apps using the user's credentials via an app installed in the user's iPhone or iPad. It is a very convenient tool, although it requires a computer running macOS or Windows to install. However, it also provides an extension to the MacOS Mail app so it can automatically re-sign apps when the iPhone or iPad is in the same Wi-Fi network. This makes the 7-day re-signing requirement for free users irrelevant for most users. There is clearly interest from the iOS developer and user communities in escaping Apple's walled App Store, as AltStore had over 1 million downloads by October of 2020 [197] and is used to distribute applications from multiple developers.

These were the steps necessary until iOS 15, however, since iOS 16 there is an extra step: turning on 'Developer Mode' (required only once). This restarts the device and requires introducing the device's password after a pop up warns users that their "device's security will be reduced". After this, AltStore can be installed and used as explained above. Thus, this change serves to add friction before the sideloading is properly activated, and also to warn users against potential dangers, in case they are being led by others to sideload applications onto their devices without knowing the risks.

Sideloadly

Sideloadly is another popular tool, it also does not require a Jailbreak, can be used in conjunction with a free or paid Apple Developer account and supports iOS 7 to iOS 18.[14] It requires a Windows or macOS computer. Because it imitates the process developers would follow for testing using Apple's Xcode, it can install apps and prevent them from expiring via cable or Wi-Fi. It also has unique features like installing unsupported iOS apps to Apple Silicon Macs. It supports sideloading apps to Apple TV.

TrollStore

TrollStore 2 is similar to the previous options but enables sideloading directly from compatible iOS devices without the need for a computer.[15] CoreTrust verifies binaries are properly signed and not malformed. TrollStore 2 exploits software bugs in older versions of CoreTrust,[16][17] in order to allow

[13] https://altstore.io
[14] https://sideloadly.io
[15] https://trollstore.app
[16] https://nvd.nist.gov/vuln/detail/cve-2022-26766
[17] https://nvd.nist.gov/vuln/detail/cve-2023-41991

sideloading, permanent signing and automatic refreshing and resigning of apps on-device. It is based on CoreTrust's improper validation of signatures allowing the reuse of App Store binary signatures to double-sign apps, bypassing signature validation. It is compatible with iOS (and iPadOS) 15.6 to 16.5 on all devices, 16.5 to 16.6.1 on A8 to A11 chip devices (iPhone 6 to iPhone X, inclusive), and iOS 15.7.7–15.8, iOS 16.5.1–16.6.1, and iOS 17.0 on A12 devices (iPhone XS) and newer due to the presence of the CoreTrust bug. TrollStore (v1) is compatible with iOS 14.0 to 15.4.1.

Enterprise certificates

However, there are alternatives to AltStore and similar self-signature-based methods. The most common of these requires trusting the enterprise certificate of a particular third-party market to then install apps from their website. Furthermore, very frequently these markets provide their own market app. These market apps are sometimes required for users wanting to download apps from markets, other times they are basically web apps with both options or only the website available to users. A market app potentially makes it easier for users to keep sideloaded apps updated.

This market certificate method adds the convenience that users do not need to install AltStore nor re-sign apps every 7 days and be limited to 10 app IDs per week (for free accounts). However, the market signature itself may be discovered by Apple and revoked, causing all apps signed with that signature to stop working for all users. Many of these markets offer subscription services in order to fund the cost of creating such accounts, and offer users guarantees of faster key rotations or lower chances of loss of service than their free accounts. Others also lift any restrictions on the number of apps downloaded per time period, or downloaded per device at any given time.

Thus, all the apps available in these markets will have been signed using the (or one of the) markets' enterprise certificates. Some of these offer the option to (1) download the already-signed apps, or (2) download the raw IPA files directly. The latter allows users to sign apps themselves using tools such as AltStore or TrollStore without needing to trust the market certificate. These aspects are discussed in detail in Chapter 4 (see §4.3.3).

Table 2.1 summarises the different certificate types available for all sideloading users. They have the following requirements and limitations:

- Enterprise certificates require phoning Apple and proving the legitimacy of the company and its use of the apps, can install unlimited apps on unlimited devices.
- Developer certificates require re-signing the apps every 365 days, can install unlimited apps in up to 100 devices (UDIDs).
- Free developer certificates require re-signing apps every 7 days, can install up to 10 apps in three of their own devices.

Of course for those with devices compatible with TrollStore and similar solutions these are more powerful than the certificates below, as they are more permanent solutions. Paid developer certificates

Table 2.1 Overview of signing certificate types.

Certificate Type	Cost (USD)	Target Audience	No. of devices	No. of apps
Enterprise certificate	299/year	Companies	Unlimited	Unlimited
Developer certificate	99/year	Developers	Up to 100	Unlimited
Free developer certificate	0	Students, future devs	3	10

are sometimes distributed by markets or individual users for money, since 100 UDIDs can be associated with each (costing USD 99/year). Many markets use enterprise certificates (see §4.3.3) and have paid tiers (see §4.4.1) that promise users a quick turnover in case a certificate gets deactivated by Apple [220, 224].

Mobile Device Management (MDM) Profiles

This is an even more invasive technique than those presented above. It is meant for organisations and companies to control company-owned devices given to their employees. They are also used in schools with a slightly different manager dashboard.[18]

The manager has many commands available, including the ability to install apps remotely to the devices, other profiles, changing configurations and logging the user out, shutting down or restart the device, querying the list of installed apps, etc.[19] They can also query user devices for hardware serial number, Unique Device Identifier, Wi-Fi, media access control (MAC) address and more.[20]

Some of the markets studied employ these as an installation method, this allows users to queue apps to be installed from different devices e.g. from a computer to be installed on their iPhones. However, it requires an incredible amount of trust on the market operators to not abuse the features of MDM profiles, as we discuss in Section 4.5.

2.4 Legal impositions on Apple by the EU and the US

While Google was ruled to have "built and maintained an illegal monopoly" with its Google search engine [123], it has recently been allowed to continue operating Android as before [123, 156]. Apple, however, faces increasing pressure from both the EU and US legal systems for its closed ecosystem: the European Commission fined Apple for "abusing its dominant position [...] through its App Store" [76], and the US Justice Department lawsuit filed in 2024 states that "Apple illegally maintains a monopoly over smartphones" [210]. The EU designated Apple as a 'gatekeeper' under their Digital Markets Act (DMA) in September 2023 [32, 79]. Under the DMA, gatekeepers have interoperability requirements in Europe including opening up NFC payments to third-parties (instead of locking users

[18] https://support.apple.com/guide/deployment/depc0aadd3fe
[19] https://support.apple.com/guide/deployment/dep789n2k1qp
[20] https://support.apple.com/guide/deployment/dep1d7afa557

to Apple Pay), allowing alternative browser engines (instead of requiring all browser apps to use Apple's WebKit), and connectivity to third-party smartwatches, headphones, etc. (instead of keeping exclusive features for their Apple Watch and AirPods) [27, 31, 78].

Most relevant for our study is the imposition on Apple to open their ecosystem to third-party app markets in the EU. Apple complied with these requirements in March 2024 with iOS 17.4, which enabled third-party app stores [31]. However, they introduced multiple fees to third-party markets and developers using them, which has been heavily criticised by competitors and hailed a 'show of compliance' or 'malicious compliance'. Apple introduced the following fees:

- 'Core Technology Fee' (CTF) of EUR 0.50 charged to developers for every first annual install after one million [26]. So updates are free within a year, but developers are charged yearly per user and device over one million.
- 'Core Technology Fee' (CTF) of EUR 0.50 is charged to alternative app marketplaces for every first annual install of their marketplace app, including those under one million. For example, they are charged for each new install by a user within a year, with free updates for a year, but charged again if they install the app on a second device and charged again yearly.
- In-app purchase (IAP) commissions have been changed in the EU where developers can continue to pay Apple the 30% commission as the rest of the world, or accept the new terms that include: requirement to pay the CTF as described above even for apps in the App Store, reduced IAP commissions of 10–17%, plus a 3% fee if using App Store payment but developers can choose an alternative payment service provider to skip this 3% fee.[21]

With these fees Apple still obtains part of the revenue, and most importantly, has kept control of the apps published for iOS both in the App Store and in third-party markets. Apple still notarises and signs all apps [27], including those that are not allowed in their App Store like apps focused on porn [218].

Many parties have criticised these fees, as they disincentivise the creation of third-party markets and developers from hosting their apps in them. This is shown by the lack of alternative markets beyond the few discussed in §2.4.1. For these reasons, Apple has been fined with EUR 500 million in the EU for non-compliance with previous mandates under the DMA. The commission finds that Apple did not freely allow developers to inform users in their App Store apps of alternative offers outside the App Store. They also found Apple imposed too many restrictions on developers to distribute their apps through alternative channels [77]. In addition to the fine, the EU ordered Apple to remove these restrictions and change their approach. This is an ongoing legal process during which Apple has been repeatedly found to be in breach of the DMA legislation, the European Commission found Apple imposed restrictions until April 2025 that meant "developers cannot fully benefit from the advantages of alternative distribution channels outside the App Store" and consumers "cannot fully

[21] https://developer.apple.com/support/apps-using-alternative-payment-providers-in-the-eu/

benefit from alternative and cheaper offers", while Apple "failed to demonstrate that these restrictions are objectively necessary and proportionate" [77].

Apple was also fined over EUR 1.8 billion in 2024 for "abusing its dominant position on the market" to give Apple Music an unfair advantage over other music streaming services, which it prevented from informing users of cheaper prices when subscribing outside the app (due to Apple's 30% commission on IAPs) [76].

Apple has had similar problems in the US, where U.S. District Judge Yvonne Gonzalez ruled Apple violated her order from 2021 to reform its App Store which had been imposed during an antitrust lawsuit initiated by Epic Games [209]. Apple had continued to charge a 27% commission on out-of-app purchases instead of its 30% for in-app-purchases, maintaining its anticompetitive advantage and revenue streams. The court found Apple "continued its anticompetitive conduct solely to maintain its revenue stream" [209]. This effectively made the situation even worse than it originally was, although developers could now add links to payments outside the apps. Most importantly, Apple had been mandated to allow developers to direct consumers to third-party purchasing mechanisms. However, although they technically complied, they added full-screen warnings (commonly known as scare screens) to dissuade customers from using these alternative payment methods. Apple was ordered to immediately remove these frictions and stop charging these unlawful fees. These changes to payments are US-only.

In 2024 the Justice Department together with the Attorneys General of 20 states sued them for their monopolisation of the US smartphone market.[22][23] This antitrust case is slower than its EU counterparts, with no results expected in the near future [55].

Apple's new CTC fees from January 2026

As a response to the EU's DMA and further investigations into the CTF fees described at the beginning of this section, Apple has introduced a Core Technology Commission fee that is similar in spirit to the CTF fees [26, 28]. While it will result in slightly lower fees for some developers according to Apple, this change has been accused of being simply more malicious compliance by Apple [163].

2.4.1 Post-EU regulation iOS sideloading options

As mentioned above, until Apple follows the EU's latest mandate and removes these fees [59], the Core Technology Fee of EUR 0.50 is paid by developers to Apple for each install after 1 million, with some caveats, and alternative app markets for each install of their market app even before the 1 million threshold [26]. Thus, there are not as many new markets as everyone expected in the EU. Also, third-party markets need to be approved by Apple and meet some criteria [23].

[22] https://www.justice.gov/archives/opa/pr/justice-department-sues-apple-monopolizing-smartphone-markets

[23] https://www.justice.gov/archives/opa/pr/four-additional-states-join-justice-departments-suit-against-apple-monopolizing-smartphone

Effects on iOS sideloading techniques and emerging markets

The situation of AltStore is now very confusing, it became the first iOS sideloading market in the EU with the release of AltStore PAL. Meanwhile, the rules around emulators in the App Store also changed, so their Delta emulator is now available in the App Store, but only in non-EU countries because Apple does not allow the distribution of apps in both sideloaded markets and the App Store at the same time. Thus, Delta is now a free app outside Europe and needs a subscription to AltStore PAL to be sideloaded in the EU due to Apple's Core Technology Fee. Furthermore, AltStore continues to be a self-signing sideloading technique available within and outside the EU.

Other markets include Epic Games (soon coming to the UK and Japan as well),[24] Aptoide (a well-known sideloading market in Android),[25] and there is also SetApp Mobile with a subscription model, and some B2B (business-to-business) solutions.[26]

Effects on the modded markets

This affects markets registered in the EU under Apple's programme, but not the modded markets studied here due to the proliferation of piracy and unauthorised modifications of apps.

However, the market AppDB still tried unsuccessfully to become a part of Apple's programme, and instead got their related developer account revoked [12]. Instead, they started transitioning their market into a third-party market with their own APIs and developer verification during the course of our study [16]. They claim they do not take any commissions from in-app purchases [14], although they changed from a free site to a subscription model [18]. Seemingly they are currently collaborating with EU authorities looking into Apple's compliance [10].

2.5 Summary

In the first days of computing, users, programmers, and sometimes even designers and builders were the same person or group. Later, computer programmes used to come included with the first business computers, after a while software and hardware were decoupled. This allowed reusing software and more general solutions to emerge, and software became its own industry. Soon punched cards were substituted by tape, these by cassettes, then floppy discs, CD-ROMs and DVDs and finally software distribution became mainly digital over the internet. This progressively made licences and DRM solutions more and more prevalent to prevent illegal copies by users and third parties. Better, faster, smaller storage solutions combined with the smaller form factor of handsets, better battery life, and better connectivity through 2G and then 3G gave rise to the smartphone, mainly with the introduction of the iPhone in 2007.

[24] https://store.epicgames.com/mobile/ios
[25] https://en.aptoide.com/ios
[26] https://macpaw.com/news/setapp-mobile-open-beta

Mobile devices and apps are one of the main ways users interact with computers nowadays, and over 4 billion people have smartphones [5]. Apple and Google created lucrative markets for app developers while reaping a portion of the benefits too. Thus, it marked one of the first times that closed markets emerged as the default software distribution method for personal computing devices. Users could sideload apps on Android, while it was a cumbersome, unofficial process on iOS.

Apple and Google's practices have been described as 'monopolistic' and have brought them under scrutiny recently. However, Apple's show of compliance and outright non-compliance in some areas have angered both the European and US justice systems and they now face further scrutiny. However, all of this has translated into very limited changes for consumers, with only a few new iOS app markets emerging in the EU more than a year after the DMA legislation was enacted. The following chapters will explore how sideloading has evolved during the last few years, and whether users are now more or less secure than before.

In terms of security, the official mobile app markets with their app review model function in a similar way as the physical stores used to do, by offering users trust that what they are downloading comes from the developers listed and has been reviewed. This is in contrast to the viruses passed around in illegal copies in floppy discs back in the day, or potential malware in sideloading markets and forums nowadays.

Chapter 3

Sideloading and modded apps in Android

This chapter covers our research into modded markets and sideloading incentives in Android. 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.

This chapter answers the following research questions with a focus on Android modded apps, thus, RQa1–RQa3 relate to our main research questions RQ1–RQ3, respectively:

RQa1 What is the ecosystem of modded Android markets and apps composed of, and what is its size?

RQa2 What are the financial incentives for operating modded Android app markets? How does this affect the original developers and markets?

RQa3 What are the security implications of installing apps from these Android markets?

Section 3.1 gives an overview of sideloading on Android and our findings. Section 3.2 details our methodology (finding and ranking modded markets, nomenclature, dataset collection and scraping, static analysis) and ethical considerations. Section 3.3 answers RQa1 by presenting the static analysis results and a case study of 10 popular apps. Section 3.4 tackles RQa2 by presenting our findings on in-market and in-app advertising and piracy. Section 3.5 answers RQa3 with a summary of our VirusTotal analysis results and dangerous changed permissions in apps with modded code. Finally,

the limitations are explored in Section 3.6, followed by a literature review (§3.7) and a summary of our findings (§3.8).

In summary, this chapter makes the following contributions:

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

3.1 Motivation

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 (including China and North Korea), or does not allow charging for apps and IAPs (In-App Purchases) (Cuba, Russia and Belarus) [95, 102, 103, 135].There are also 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 provide interfaces similar to Google Play and third-party markets, but focus on or advertise a large catalogue of predominantly modded apps. Modded apps may (for free): bypass or 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, while modded markets do not report numbers of downloads, we calculated the lifetime revenue of these apps in the official Google Play. These paid apps originally have a total value of USD $33 975 in Google Play and an estimated lifetime revenue in Google Play (price $\times$ Google Play installs) of around $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 [109]. These numbers are merely presented here to illustrate the size of these 5% of modded apps in the official markets. However, they do not reflect the income reduction suffered by the developers (see §3.4.4 for further discussion).

Furthermore, 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 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. 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) in relation to the Digital Markets Act [80, 81]. The UK's Competition and Markets Authority (CMA) has also investigated these questions [203].

3.2 Methodology

This section introduces the methodology and nomenclature used in this chapter. Some of it is common to other chapters and will be reiterated as needed.

3.2.1 Identifying and ranking modded Android 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/Truecaller} mod'. The complete list of keywords is provided in Appendix A.1.1. 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 of the market names' popularity 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 [166]. 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.

3.2.2 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 in 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.

3.2.3 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 [2]. We scraped the 13 most popular modded markets (see §3.2.1) 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 [179, 215].

We make the ModZoo dataset available to the research community (§1.4).

3.2.4 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 §3.2.2). It then analyses the obtained AndroZoo match and stores the results. We run the third-party reverse engineering tool Apktool [114] 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, from which we 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 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.

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).

3.2.5 Ethical considerations

Apps were only collected for analysis, not used, except for the case study of 28 modded apps see §3.3.2). 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, minimising any additional load placed on them. 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.

3.3 The Android modded app ecosystem

This section tackles RQa1: "What is the ecosystem of modded Android markets and apps composed of, 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 most popular modded app markets.

3.3.1 Analysis of modded apps and markets

Our technical analysis focused on the 13 highest-ranked modded markets, as determined in §3.2.1. Their average estimated size based on number of apps listed is 37 486 apps with a mean of 15 719 apps

Table 3.1 Overview of the modded market ecosystem and proportion of modded apps.
['Est. Size'=Estimated Size, 'Unique Packs.'=Unique Packages, 'Unch. Apps'=Unchanged Apps]

Market	Est. Size	Unique Apps	Unique Packs.	Duplicates	Paid (%)	Modded Apps	Unch. 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

downloaded (see Table 3.1). The difference is due to unavailability of some apps and broken download links. Markets marked with asterisks (∗) in Table 3.1 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 self-reported 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.

The 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, with 39 988 out of the 146 162 unique apps found in more than one market. 'Duplicates' are apps advertised as different versions within a market which turn out to be hash-identical duplicates of other apps or versions in the same market.

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

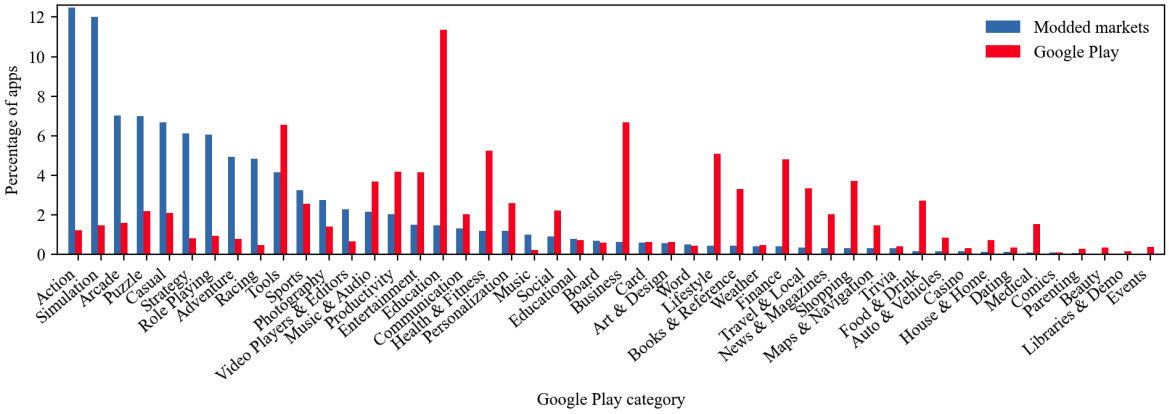


Fig. 3.1 Distribution of Google Play app categories in the modded markets and Google Play

matches, we compute the number of code-identical and code-modded apps as defined in §3.2.3: 81 250 apps (68.1%) have received changes to their code ('Modded Code' in Table 3.1). Code-modded apps are closely related to permission-modded apps, as discussed later (see §3.3.3). 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.

App categories in modded markets and Google Play

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 3.1, 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. 'Education' is the most common category in Google Play, but only the 17th most common in modded markets. Education apps include "exam preparations, study-aids, vocabulary, educational games, language learning" according to Google [98]. These apps are mainly non-competitive puzzle games, language learning apps, and apps made for children. It is possible that there is low demand for these apps on modded markets because parents and users do not want to install these apps from unofficial markets due to their important educational nature and their intended audience. Education should not be confused with 'Educational', a separate game category that is much less prevalent in Google Play, as shown in Figure 3.1.

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 [52]: 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.

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. 3.2, 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 is $2.28 billion. We obtained this as the product of their current price in the US Google Play times the number of installs as reported by Google (US price $\times$ number of global installs in Google Play). This estimate excludes any possible in-app purchases. All modded markets studied lack payment mechanisms. Thus, the paid apps in Table 3.1 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).

Based on their US Google Play prices, 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 each market had to buy each of the paid apps they host individually (not counting multiple versions, only unique app packages). It is possible market operators downloaded paid apps and requested refunds after making copies [101], resulting in $0 of revenue for the original developers. As mentioned before and discussed in §3.4.4, the lost revenue is not easy to estimate, and these numbers are presented to illustrate the size of the problem even though modded markets do not disclose the number of downloads of the apps they host.

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. The maximum, minimum and mean IAPs prices (in Google Play) are shown in Fig. 3.2. It shows most paid apps cost well under $20 and the majority of IAPs have low minimum prices. However, mean IAP prices are over $25 for almost 50% of the apps, and over 40% of the apps have IAPs with prices over $100 in Google Play (shown in the maximum IAP prices). The total price of the IAPs 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 §3.3.2).

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, as mentioned in §3.3.1. Waiting periods are common, preventing users from

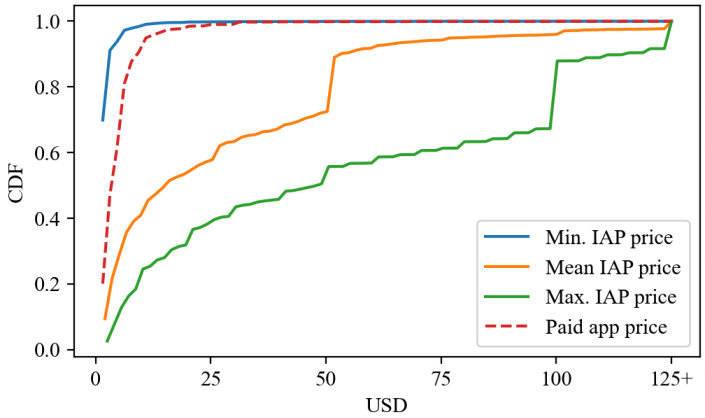


Fig. 3.2 Google Play paid apps and IAPs price CDF.

downloading multiple APKs in a short period, it provides an opportunity to show users more ads. Some markets use CAPTCHA tests, and a small number implement Cloudflare DDoS and bot protection [60].

Some markets introduced anti-scraping protections during our study. It is possible that our contacting markets for our surveys, 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. Their websites merely point users to their market app download link. Months into the study HappyMod started redirecting users to Jojoy. Our scrapers were unaffected as they obtain the metadata from the websites and contact the shared endpoints to download the APKs directly without using the market apps.

3.3.2 Case study

The ten 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 Google Play versions alongside some modded versions of each app from different markets to analyse their modded features and assess the scale of potential 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 simply click ‘Install anyway’.

Many modded apps showed a small logo or pop-up with the market’s and sometimes also the 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 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, the other 12 versions had unchanged ad IDs.

TikTok

A big part of TikTok's revenue comes from IAPs in the form of coins users can send to their favourite creators during livestreams, triggering an animation and resulting in revenue for creators. TikTok reported $1.5 billion IAP revenue in 2022 [130, 189]. These coins come in bundles costing USD $0.07–249.00 for 5–17 500 coins. Modded versions claimed to offer unlimited coins, downloads without watermarks and geolocation restrictions removed. Downloading without watermarks worked well, but coins were not included in any versions we tried.

SHAREit

SHAREit offers premium features for $1.99/month, including removing ads, exclusive customer service, and regular cleanup and antivirus. Modded versions of this app claim to remove all ads and include all premium features. None of the versions we tested removed all ads, only one provided regular cleanups and none provided premium customer service.

Telegram

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

Spotify

Spotify Premium costs $9.99/month and provides an ad-free experience, 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 [90]. They reported a €11.57 billion revenue from Premium subscriptions and €1.7 billion ad revenue from non-premium users in 2023 [109]. Modded versions advertise having the premium features. They were ad-free, provided unlimited skips and the ability to play any song and playing them in any order. However, none can provide downloads or higher quality audio, nor the ability to select a different device to play on. Not all modded markets make it clear they cannot provide these features.

Truecaller

Truecaller Premium costs $4.99/month or $49.99/year and includes 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.

Subway Surfers

The game Subway Surfers offers different 'coins' and 'keys' bundles for $0.99–99.99. Modded versions claim to have all these IAPs unlocked, some even offer 'God mode' game-play advantages: unlimited jumps, flying, etc. Piracy cost this game $91 million by 2018 according to Tapcore [128, 129]. The versions we tested provided free IAPs and unlimited coins.

Candy Crush Saga

The game Candy Crush Saga offers many perks bundles from $0.99–99.99 and 'gold' for $1.99–99.99. All modded markets advertise having all levels unlocked, infinite lives, boosters, etc. Such offerings render gold, lives, and other IAPs useless. The modded versions tested worked as advertised.

Free Fire

The game Free Fire offers subscriptions for weapons and perks costing $7.99–12.99/month, and 'diamonds' bundles for $0.99–49.99. Modded markets advertise unlimited money and diamonds, as well as gameplay-related mods including aim-assist, no recoil, and hacks [213]. Most versions tested did not work and the rest had none of these features.

My Talking Tom

The game My Talking Tom offers 'diamonds' for $1.99–99.99, and monthly subscriptions for perks at $4.99/month. A purchase is required to remove all ads. Modded apps offer unlimited coins, perks and no ads. From our testing unlimited coins unlock most content, but some ads including full-screen pop-up ads are still present and subscription perks remained locked.

Hill Climb Racing

The game Hill Climb Racing sells perks and coins in multiple bundles for $1.99–59.99. Unlocking all vehicles costs $29.99, all levels $29.99, and unlocking everything $54.99. Specific bundles also remove ads when bought. Modded versions advertise having all content unlocked or including unlimited coins. All versions tested provide unlimited coins, letting users unlock all content, although with banner and pop-up ads still present.

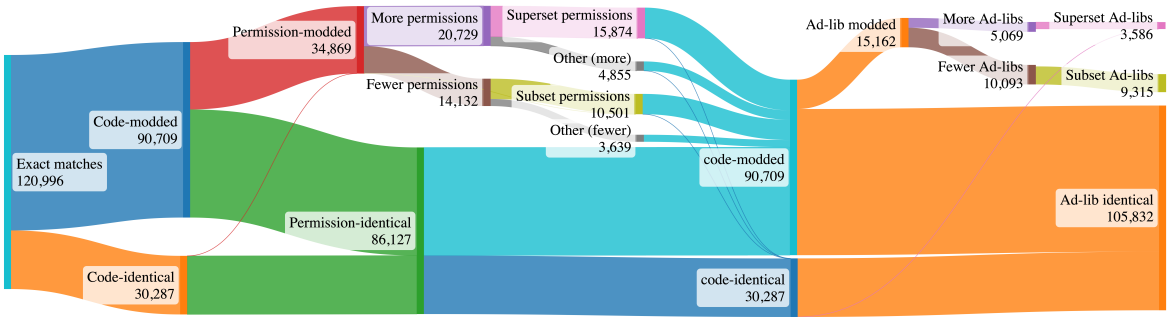


Fig. 3.3 Permissions and ad library changes in code-identical and code-modded apps.

3.3.3 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. 3.3). 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 ad libraries. 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 also hypothesise permissions are sometimes added to code-modded apps in order to increase ad revenue for the modder, which would also affect users' privacy. 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 §3.3.1). Unmodified apps may be useful to users with older devices for compatibility or features.

3.3.4 App signing certificates

Most markets use mainly debugging and default Android Studio certificates unfit for app publishing. There are 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.

RQa1: What is the ecosystem of modded Android markets and apps composed of, and what is its size?

The analysed modded markets typically have tens of thousands of apps each which we downloaded. The biggest has an estimated size of over 220k 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.

3.4 Market operator motivations and income

This section tackles RQa2: "What are the financial incentives for operating modded Android 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.

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

3.4.2 Advertising libraries and advertiser IDs

Many code-modded apps are advertised as ad-free versions of 'Freemium' apps. We use a safelist (see §3.2.4) 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 code-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.

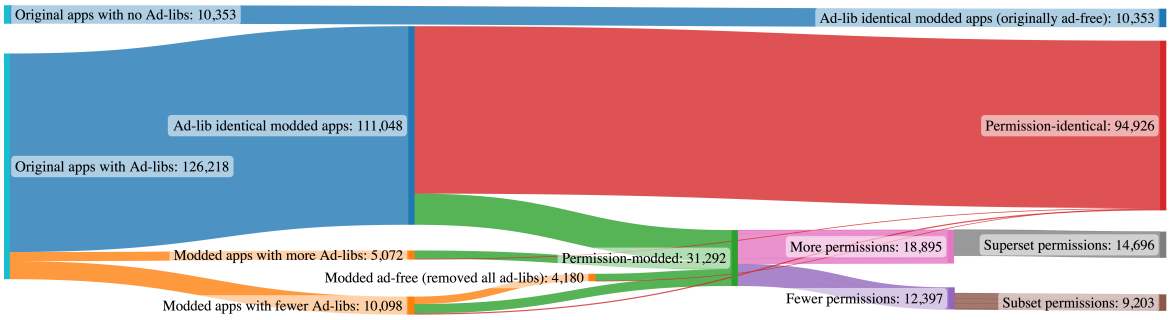


Fig. 3.4 Distribution of ad libraries in original and modded apps and modded permissions.

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.

3.4.3 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. 3.4. Thus, there is a genuine small offering of ad-free versions of popular apps with smaller permissions sets. Furthermore, as shown in Fig. 3.4, 88.9% of modded apps are ad-library-identical to their Google Play counterparts when counting those without ad libraries too, or 88.0% of those with ad libraries. 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,

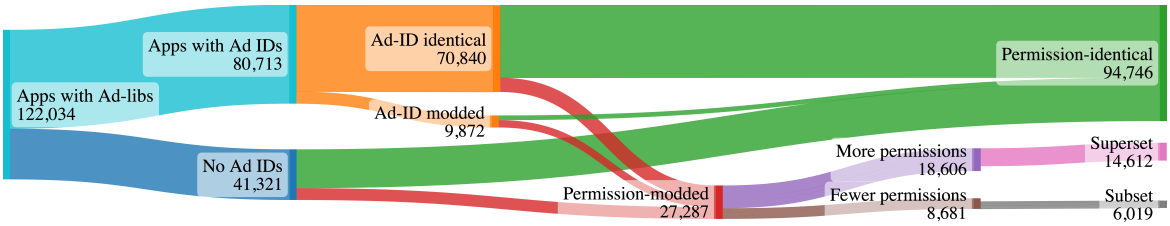


Fig. 3.5 Distribution of permissions and advertiser IDs in modded apps with ad libraries.

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. 3.5. 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%).

These results suggest a strong correlation between permissions and advertising modifications: 91.3% of apps with added ad libraries are permission-modded, 64.9% with added permissions; 84.3% of apps with unchanged ad libraries are permission-identical; and 61.1% of ad-ID-modded apps are permission-modded. 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.

3.4.4 Modded apps, piracy and user displacement

Piracy has been found to have a negative impact on revenue in the music and media industries [174, 188]. Tapcore estimated 14 billion app installs were pirated in 2017, costing app developers $3-4 billion annually [128]. They also estimated pirated apps had cost developers $17.5 billion in revenue between 2013 and 2018 [129], and Subway Surfers $91 million. It is difficult to estimate the current revenue loss with the growth of IAPs and ad revenue in apps and games since 2018. Furthermore, these figures must be considered with reserve, since they come from a company affected by piracy and it is unclear how they were calculated.

Ubisoft's CEO and the creators of Game Dev Tycoon estimated over 90% of players had pirated their games in the early 2010s [46]. CD-Projekt RED estimated 3–4 times as many pirated copies as legitimate ones existed of their hit game series "The Witcher" [46]. Others made lower estimates of 14.6% of global PC game downloads being illegal [46]. Again, these are estimates by affected companies and should be taken with caution.

A survey of 50 742 PC gamers from dozens of countries found 35.8% of respondents currently pirate games [85]. Studies on computer game piracy found high displacement rates of -2.49 for games, meaning each illegal download of a game typically displaces (prevents) multiple genuine purchases [167]. Unlike for music, films, series, and books, 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. Although all economic models and estimations have uncertainties, this suggests game piracy results in user displacement and loss of revenue.

RQa2: What are the financial incentives for operating modded Android app markets? How does this affect the original developers and markets?

We identified possible revenue streams for market operators and modders including ads and sponsored posts (§3.4.1). One in five code-modded apps with ad IDs had them changed (§3.4.2). Furthermore, our analysis points to 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 mostly permission and ad-library-identical (§3.4.3). We found 6 984 pirated apps with $2.28 billion in estimated lifetime revenue in Google Play (official price times number of downloads on Google Play), and 100k apps with IAPs typically offered for free in modded apps (see §3.3.1–§3.3.2). Modded apps may cause user displacement, disrupting original developers' and markets' revenue and innovation (§3.4.4).

3.5 Security implications of modded apps and markets

This section answers RQa3: "What are the security implications of installing apps from these Android 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 §3.3.2). This has security and economic implications. 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 or change what a user can store without being detected. It may also affect other Internet users, e.g. if modded apps embed a botnet.

3.5.1 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 [179, 215]. Similarly, the recommended threshold of around 10% of antivirus engines (AVs) flagging APKs as malicious is used (see §3.7). Furthermore, existing [184] 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 (see §3.6).

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 [2].

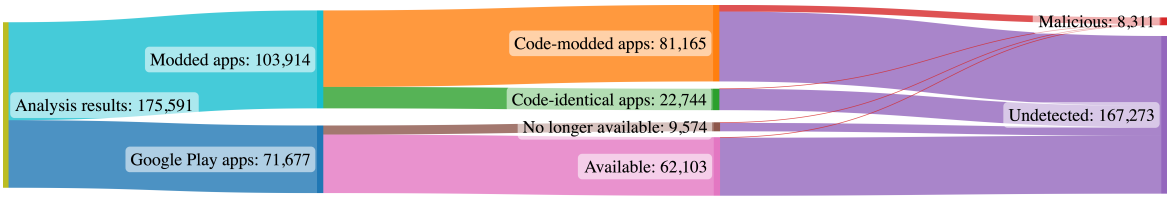


Fig. 3.6 Distribution of malicious apps across Google Play and modded apps.

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 3.6. 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: 13.4% 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 32 most prominent labels are shown in Table 3.2. Four of the top 20 are not present in Google Play apps at all, while most others have a significantly bigger presence in modded apps.

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 3.3, 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

Table 3.2 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
appflood	11	8	3
igexin	10	7	3
virtualapp	7	5	2
mcalprotect	6	3	3
remco	6	3	3
revpn	6	5	1
agentsmith	5	4	1
apkprotector	5	5	0
clicker	5	4	1
miniupnp	5	3	2
subspod	5	0	5
utilcode	5	3	2

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' [97]. Some permissions are classified as "Not for use by third-party applications", we categorise them as very dangerous in Table 3.3, e.g. 'READ_LOGS' .

Signature permissions can only be used by apps signed with the same certificate as the app that declared it. Thus, many of these are usually reserved for system apps and similar or apps by the same developer sharing functionality or data. However, some signature permissions like 'SYSTEM_ALTER_WINDOW' and 'WRITE_SETTINGS' have 'appop' (App Operation) and 'pre23' as part of their protection level as well. This means they are granted to apps downloaded from Google Play automatically (unless they target API Level 23 or higher), and can be activated manually by users for apps targeting API Level 23 or higher and sideloaded apps [97]. These have been abused in malware, phishing, and adware apps [67, 87, 192]. Thus, Google recently introduced warnings on Android 13 [182] and expanded them on Android 15 [170], making it more difficult for users to manually grant dangerous permissions to sideloaded apps. The restricted permissions include our system permissions: 'SYSTEM_ALERT_WINDOW', 'REQUEST_INSTALL_PACKAGES', 'WRITE_SETTINGS', among others [96]. These are still dangerous for users with older devices and versions of Android. 'BATTERY_STATS' is unusable by sideloaded apps as it lacks the 'appop' protection level, and is a signature and privileged permission [97].

Finally, some permissions are not categorised by Google even though they are relatively popular in both Google Play and modded apps. We classify 'android.permission.READ_SETTINGS', the only example in Table 3.3, as 'uncategorised'.

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.

Table 3.3 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

RQa3: What are the security implications of installing apps from these Android markets?

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. These affect users' privacy and can make them vulnerable to phishing and malware that abuses accessibility features. Newer versions of Android have restricted many of these dangerous permissions, but users with older versions are still vulnerable.

3.6 Limitations and threats to validity

The ModZoo dataset is potentially biased if the modded markets list is missing important markets. However, we cross referenced our ranking with the Tranco ranking, confirming we included all the top modded markets as measured by Google Trends and the Tranco ranking in the months leading up to our study. 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 [38, 141, 217, 223], They are infeasible considering the scale of ModZoo as they require downloading all libraries of interest and their execution time typically increases exponentially with the number of libraries of interest and apps analysed. A safelist is an acceptable compromise between accuracy, speed and scale since code obfuscation and shrinking are not enabled by default in Android Studio [93]. Google Play reports IAP prices as ranges instead of their number, price and which are subscriptions. This introduced imprecision in the study of IAPs at scale. IAPs were found not to cover the full price range reported by Google, further showing their inaccuracy.

3.7 Literature review

Wang et al. analysed 6 million Android apps in 16 Chinese markets, inter-market similarity, publishing behaviours, malicious and fake apps [214]. 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 [110]. 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 [115] and firmware over-the-air [47] 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 [134]. Kumar et al. analysed differences in 26 countries' Google Play markets, finding apps are often unavailable due to developer-introduced geoblocking [131]. 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 [185]. We found the opposite for modded markets, as operators lack the motivation that 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 [124]. 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 [36], created permissions graphs to find outliers [190, 194], and found malware-related permissions based on other datasets [3]. 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 [136, 193], while some identified packages and APIs used [1]. 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 [225]. 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 [225]. 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 [50]. 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 [161]. Others used weighted voting, relied on supervised learning, and used future results (after 4 weeks) as ground truth [120]. Others confirmed the increased accuracy of older results [178, 179]. Others focused on native code libraries misuse in a small sample of Android apps from one third-party market [216]. 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 [58]. 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 [92], 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.

A very recent study from 2025 surveyed 20 modded WhatsApp users in Kenya, who used the mod to hide their behaviour and stalk others [153]. Some users incorrectly credited features as mod-exclusive when they are included in the official version of the app. Worryingly, almost half of the participants trusted the modded versions more than the official one despite the excessive permissions and the presence of malware in some of them.

3.8 Summary

This chapter 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 over 280k apps is available to researchers (§1.4). 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.

Chapter 4

Sideloading and modded apps in iOS

This chapter presents our research on iOS sideloading and modded apps. Apple's mobile ecosystem is often considered a 'walled garden' where sideloading of apps is not possible without a jailbroken (rooted) iPhone. This is not the case (see §2.3). We found 89 third-party app markets and present the first overview of the modded app ecosystem on iOS. We collected and analysed apps from the 9 most popular modded markets over 11 months. About 28.9% of the modded apps are pirated paid apps offered for free with a total of 1.12 billion self-reported downloads across just three markets. This affects developer revenue. Furthermore, many modded apps offer subscription features and in-app or in-game items for free: 69% of modded apps originally had in-app purchases in the App Store. In terms of user security, VirusTotal classified over 1% of modded apps as malicious (CVE, exploits, trojans mainly), versus less than 0.03% of App Store apps. Unlike Android, all iOS apps must be signed by an Apple-approved certificate, including sideloaded apps (§2.3). Apple checks these certificates before apps are first run and periodically. Modded markets circumvent this by signing 78% of the modded apps with enterprise certificates. Others rely on device management profiles meant for company devices, which can be abused to extract private data and remotely install apps on users' iPhones.

The research questions this chapter answers related to iOS apps are the following. RQi1–RQi3 concern the main research questions RQ1–RQ3, RQi4 answers RQ1 further by comparing the Android and iOS modded app ecosystem:

RQi1 What is the ecosystem of modded iOS markets and apps composed of, and what is its size?

RQi2 What are the financial incentives for operating modded iOS app markets? How does this affect the original developers and markets?

RQi3 What are the security implications of installing apps from these iOS markets?

RQi4 How does the ecosystem of modded iOS apps differ from its Android counterpart?

Section 4.1 summarises this chapter's motivation. Section 4.2 details our methodology (finding and ranking modded markets, dataset collection and scraping, static analysis) and ethical considerations.

Section 4.3 answers RQi1 by presenting the static analysis results and a case study of popular apps. Section 4.4 tackles RQi2 by presenting our findings on market monetisation strategies, in-market and in-app advertising, piracy, number of downloads, and active users. Section 4.5 answers RQi3 with a summary of our VirusTotal analysis results and the security implications of the different sideloading methods. Finally, the limitations are explored in Section 4.7, followed by a literature review (§4.8) and a summary of our findings (§4.9).

This chapter makes the following contributions:

- An overview of the iOS sideloading and modded app ecosystem and the first in-depth study of markets containing modded iOS apps.
- Monitoring, data collection and analysis of the 9 most popular modded markets over an 11-month period, collecting 40 389 modded iOS apps and their metadata (*iOSModZoo* dataset).
- The collection of the equivalent apps from Apple's App Store over the same 11-month period, resulting in the *iOSZoo* dataset containing 77 189 apps and their metadata.
- The presence of these modded markets is likely to reduce income for app developers and official markets: many modded markets have premium subscriptions for access to exclusive mods and fewer revocations; 28.9% of the modded apps are pirated paid apps offered for free, they have over 1 billion downloads; almost 7 in 10 modded apps originally have IAPs, most of their features offered for free in modded versions.
- Modded apps are riskier for users: VirusTotal classified over 1% of modded apps as malicious (CVE, exploits, trojans mainly), versus less than 0.03% of App Store apps; markets sign 78% of the apps with enterprise certificates, others use device management profiles which can extract private data and remotely install apps in users' iPhones.

4.1 Motivation

Sideloading has always been possible in consumer laptop and desktop operating systems, it then became unusual with the introduction of smartphones and their app stores. Sideloading gives users more choice and allows developers to sign and sell their own apps directly to users. Developers can also offer users extra features and subscriptions without paying a percentage of revenue to monopolist markets. However, our work shows that modded apps have significant negative effects as they are typically unauthorised modifications of apps available in official markets. While third-party markets have the potential to benefit users and developers, we argue that stricter regulation is required to protect users and developer revenue streams. Our findings point to a growing community of marketplace operators, modders and a steady user base. This work is timely for regulators, who need to balance competition and fair markets with intellectual property and user 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) [80, 81].

4.2 Methodology

We have found 89 sideloading and modded iOS app markets. Some of them, such as 'Malavida', 'Platinmods', 'TopStore', 'PDALIFE', and 'TutuApp' were already studied in the context of Android modded apps in Chapter 3. We rank the markets found and scrape the most popular 9 over an 11-month period between 1 December 2023 and 30 October 2024. The resulting dataset of 40 389 modded iOS apps, *iOSModZoo* gives us a unique account of the modded iOS app ecosystem. We analyse the apps in iOSModZoo and collect their App Store counterparts over the same 11-month period, resulting in the *iOSZoo* dataset. iOSZoo consists of 77 189 apps and, like iOSModZoo, is available to other researchers.

4.2.1 Identifying and ranking modded markets

We follow the same approach and keyword list used for our ModZoo study described in Chapter 3 but adapted to iOS (see the keyword list in Appendix A.1.2) to obtain a list of 99 iOS sideloading and modded app markets.

All 99 markets cannot be scraped and analysed in depth, thus a popularity-based ranking of the markets was curated based on the Tranco list [166]. We manually verified the markets contained apps marketed as modded, and found some of the markets only contained old app versions, had changed focus to only Android, simply no longer worked, or required paid subscriptions. We ranked the markets based on their Tranco ranking corresponding to the 5-month period leading up to our study [166]. We scraped the top 10 markets: Mobilism (* see below in §4.2.2), Platinmods, PDALife, iOSGods, AppCake, AppDB, IPAOMTK, Malavida, TopStore and Zeus.

4.2.2 Problems encountered and exceptions

While we tried our best to scrape the 10 most popular markets (see full list in Appendix A.2), we encountered some problems.

Firstly, we only managed to scrape Mobilism's metadata because all their apps are hosted in third-party services that require solving CAPTCHA tests to download the apps. While this makes the market potentially a lot cheaper to run and more resilient to copyright claims, it made it impossible for us to scrape the market at scale beyond its metadata. Mobilism's metadata includes the number of users and views and comments in each post (app version). Typically, the first post in a thread introduces a new modded app version. Mobilism works as a forum more than an app market. We tried to include some of their apps in our case study to compensate for their absence in our dataset. However, we could not add them since the versions they hosted of the case study apps were very old: WhatsApp and Spotify versions from 2018, YouTube from 2017, Instagram from 2019, and TikTok, Roblox, 8 Ball Pool, Candy Crush Saga and Sniper 3D were not even available. It seems Mobilism (the iOS side of the market at least) is past its peak in terms of user and modder engagement

(Section 4.4.3 discusses user evolution in Mobilism in more detail). The fact that it is also an Android market might have skewed its Tranco ranking.

Secondly, we made an exception with AltStore and Scarlet because they are app signing methods as well as third-party markets. AltStore is technically ranked third before PDALife. Their app signing method allows users to sign any sideloaded apps. AltStore also hosts original apps not allowed in the App Store, users can also add third-party repositories to download other approved developers' apps, and have recently become an Apple-approved third-party market in Europe (§2.3.2). Scarlet (ranked before iOSGods) is a similar case of a market used primarily to sign other markets' apps, it hosts only 6 modded apps and allows adding third-party repositories. Some of these repositories may include modded apps, but these markets are not big modded app markets.

Thirdly, other markets had to be skipped due to their unavailability: Panda Helper comes between Scarlet and iOSGods (ranked fourth), and TutuApp comes before IPAOMTK (seventh). However, Panda Helper was down for the duration of this study with invalid certificates. TutuApp advertises a VIP paid membership with access to working apps, but their free tier was unavailable for the duration of our study. This is due to the way this market operates, with presigned apps (see §2.3.2).

4.2.3 iOSModZoo dataset collection

We scraped 9 modded app markets over an 11-month period between 1 December 2023 and 30 October 2024. The resulting dataset, *iOSModZoo*, consists of 40 389 iOS apps (IPA files) from 9 of the 10 most popular iOS modded app markets, with the metadata of the remaining market (Mobilism) also included. Most apps in iOSModZoo are modded or advertised as modded apps.

Mobilism is the first market in our ranking, and we scraped all metadata and user information during the 11-month period of our study. Together with Platinmods, it is one of the few markets with user metadata. Platinmods, AppDB, AppCake and iOSGods show install numbers, and Malavida, PDALIFE, AppDB, and IPAOMTK show users' app ratings. We executed a full scrape cycle approximately every two weeks, collecting metadata pages and IPA files. To speed up the data collection process, we developed parallel scrapers using multi-threading. Our scrapers use a set of rotating dynamic proxies located in multiple geographic regions to avoid IP bans.

AppCake had CAPTCHAs embedded in almost every page, requiring manual download of apps after automated scraping of all metadata and download links. Thus, we limited our manual collection of AppCake apps to those in the 'Tweaked' section of the platform. Platinmods and AppDB required emulating button clicks and navigation using Selenium and Mozilla's Gecko Driver. Topstore and iOSGods required us to obtain a session cookie by manually signing in every few hours, which was then used to complete the automated scraping. Finally, Platinmods, AppCake and AppDB do not host apps in their own infrastructure. They instead rely on multiple third-party file hosting services to store and distribute the apps. The most common third-party hosting services are shown in Appendix A.3. We implemented downloaders for the most popular platforms (Google Drive, MediaFire and Mega.nz) that do not have advanced technical restrictions such as CAPTCHAs, rate limiting, etc.

App packages (IPAs) are stored based on their SHA256 hash to avoid duplicates. Furthermore, we developed a storage-saving solution to take further advantage of the repeated files inside modded and official apps and their multiple versions, it is detailed in Appendix 4.2.5. Our iOSModZoo dataset is available to the research community at https://www.cambridgecybercrime.uk/datasets.html.

4.2.4 iOSZoo dataset collection

There are no existing up-to-date iOS App Store app datasets similar to AndroZoo [2], which includes over 25 million apps from Android's Google Play and smaller markets. Thus, a comparison with official App Store apps is only possible after the creation of our *iOSZoo* dataset consisting of 77 189 apps. After analysing the apps in iOSModZoo, we collected their App Store counterparts over the same 11-month period, resulting in the iOSZoo dataset. Thus, the iOSZoo dataset contains every version of every (free) app with bundle IDs present in iOSModZoo published during our study and their metadata.

Our approach is limited due to storage, compared to AndroZoo's approach of scraping as much as possible from Google's Play Store [2]. However, unlike earlier versions of AndroZoo, our iOSZoo dataset also includes the apps' metadata. As mentioned above, we developed and used a storage saving method that reduces the sizes of iOSModZoo and iOSZoo (see §4.2.5).

We scraped the US App Store daily using a third-party tool to download the apps with bundle IDs present in the iOSModZoo dataset.[1] We also collected the apps' metadata using our custom scraper to query the iTunes Lookup API.[2] Like iOSModZoo, our iOSZoo dataset is available to the research community through the Cybercrime Centre at https://www.cambridgecybercrime.uk/datasets.html.

4.2.5 Storage saving technique

Because we scraped modded markets as well as partially scraping the App Store, we employed two storage saving methods. First, we stored apps based on their hashes to only store one copy of each bit-wise unique app and more easily split them into subfolders of manageable sizes. Second, we came up with a storage saving solution based on the structure of IPAs and ZIP files. IPA files are an extension of the ZIP standard, so they have similar properties, namely they can contain compressed folder structures and files. They also contain an 'end of central directory record' located at the end of the archive. Also, each of the compressed files and folders is included in order and preceded by a 'local file header'. We leverage this information to get all the subfiles (not the folders) included in the IPA (ZIP) file and store them separately based on their hash for easy retrieval.

Thus, our solution turns each ZIP file (IPA in this project) into the following resulting files:

- A single ModZip file `{file_sha256}.modzip` which contains all the structure and subfolders (still compressed), minus the TruncZips.

[1] https://github.com/majd/ipatool
[2] https://itunes.apple.com/lookup?bundleId={bundleID}

Table 4.1 Dataset sizes with and without the ModZips technique

Dataset	Only Hashes	Hashes and ModZips	Saving
iOSModZoo	9735.01 GB	9349.34 GB	3.96%
iOSZoo	16 553.97 GB	11 600.87 GB	29.92%
Total	26 288.98 GB	20 950.21 GB	20.31%

- Multiple TruncZip files which contain each a single compressed file `{file_portion_sha256}-.trunczip`.
- Finally, a single CSV file `{file_sha256}.csv` which contains the list of all TruncZip files with their SHA256s i.e. filenames and original position in the compressed file.

It is important to note that the TruncZip files are not in a correct format to be decompressed on their own as they have the local file header but are missing an 'end of central directory record' needed for a well-formed ZIP file. They can, however, be mixed and matched according to the specification in the CSV files and included in different ModZips as needed to reconstruct the original files.

This was done based on the intuition that many of these files are present within multiple IPA files, e.g. assets in games and applications, common libraries, etc. We aimed to save storage due to the large number of versions collected for each app. Our intuition was confirmed by the storage savings achieved of over 20% as detailed below, despite our scraping only running for an 11-month period.

Assuming we had used only the hashes, the storage taken by each dataset is included in Table 4.1 and the total would have been over 26 TB, our solution saved 20.3% by using ModZips. As shown in Table 4.1 the savings are bigger in the iOSZoo dataset, this is due to the daily scraping of the App Store, and the amount of official versions of each app, which share assets and libraries.

This storage saving solution is included in an open-source repository available here: https://github.com/luisadnsaavedra/ModZip. It contains a more detailed step-by-step description and runs all deconstruction and reconstruction steps with an example app.

4.2.6 Static analysis methodology

Our analysis pipeline starts by obtaining all modded apps and rebuilding them in batches to analyse them and delete them, keeping the working directory manageable. It uses the storage saving solution code (§4.2.5) to rebuild each app to its original form. It obtains and stores the SHA256 hashes of the full app (IPA) file, its Payload and the `Payload/{app_name}.app` file. It also obtains and stores the full `Info.plist` file from the `{app_name}.app` file.

General analysis and permissions

The `Info.plist` file is equivalent to Android's Manifest file. From the `Info.plist` file we obtain the app's bundle ID, name, version, and list of permissions. The list of permissions is obtained by comparing entries in the `Info.plist` file with a list of permissions elaborated from the official

documentation (included in Appendix A.4). Although not the case for all permissions, iOS requires a description of the usage the app makes of privacy-sensitive data to be included in the `Info.plist` file. Otherwise, the app will crash on execution when trying to access the resource, including the Camera, Bluetooth, Location, Microphones, Health data, Contacts and others.[3] The exact same analysis is performed for the iOSZoo dataset (App Store apps). Here the hashes of the different files are not useful since Apple signs the apps for each user individually, unlike the modded markets.

Advertising networks and ad IDs

The `Info.plist` files also contain the Google AdMob and AppLovin advertising IDs if present, and the AdNetwork identifiers. The Google AdMob and AppLovin IDs tie advertising revenue with the original developer. They are simply included in the `Info.plist` file, similar to Android. The advertising networks, equivalent to the ad libraries in Android, are also listed in the `Info.plist` file by their ID (alphanumerical strings). We have obtained a list of the more common ad network IDs from different sources, although it might be incomplete since Apple does not publish an official list.

Certificate analysis

Apps are signed by certificates: signed by Apple for official apps downloaded from the App Store and in alternative markets in the EU; signed by companies using enterprise certificates for internal testing; or signed by individual developers to test in their own devices (see §2.3.2). The modded markets use a mix of enterprise certificates and self-signed apps that users can install via AltStore, TrollStore, etc. We analysed the modded apps' certificates (in `{app_name}.app/embedded.mobileprovision`) obtaining their 'TeamName' and expiration date. We do not analyse the iOSZoo app certificates because all App Store apps are signed by Apple. The results are discussed in §4.3.3.

Modded and App Store apps matching

The iOSZoo dataset contains all App Store apps with bundle IDs found in the iOSModZoo dataset for the duration of our study (see §4.2.4). However, the App Store only lets users download the latest app version, so not all modded apps in iOSModZoo have an exact iOSZoo counterpart. Thus, modded apps need to be matched to their exact matches if available, or the closest version number from the App Store to compare them and analyse the changes present in the modded versions.

Of the 40 389 modded apps, 6 965 exact and 17 303 closest matches were found, while 16 121 apps did not match anything on the App Store. The latter include mainly paid apps and games, Apple Arcade games, and some apps and games with bundle IDs that do not exist in the App Store, perhaps because they are region-specific, modded market exclusive apps taken down from or not allowed in the App Store (e.g. emulators, clipboard managers, pornography or piracy-related, etc.).

[3] https://developer.apple.com/library/archive/qa/qa1937/_index.html

4.2.7 VirusTotal analysis methodology

Interestingly, only 3 751 modded app hashes return results on VirusTotal (having tried for each app the full SHA256, the SHA256 of the Payload folder, and the SHA256 of the `Payload/{app_name}.app file`). This shows how rarely iOS modded market operators and users analyse apps using VirusTotal and similar antivirus tools; perhaps because markets with pre-signed apps directly install the apps on users' devices without giving them access to the IPA files.

We send all iOSModZoo and iOSZoo apps programmatically to VirusTotal to be analysed, since we are unable to leverage the few crowdsourced results unlike previous studies [176, 179, 215]. This process took weeks, and apps over 650 MB had to be divided into parts and sent separately. All the analysis results of the apps sent in multiple parts were combined to obtain a score for the entire app. We argue that the false positive rate should not be affected by this, while the false negative rate might actually have been affected by VirusTotal not getting the full file as a unit.

We used a threshold of at least three antivirus engines flagging an app or app part as malicious as an indicator the app is malicious. All apps with at least one malicious part (using threshold $T = 3$) were considered malicious. This threshold is based on recommendations from previous literature [179, 215], although smaller than the one we used in ModZoo because most antivirus engines are incompatible with IPA files. The results we obtained for all 40 389 modded (iOSModZoo) and 77 189 App Store (iOSZoo) apps are presented in §4.5.1.

4.2.8 Random sample from the App Store

We obtained a random sample of 1 million App Store apps because our iOSZoo dataset could not give us insights into paid apps in the App Store, app categories in modded markets compared to the App Store, etc. This metadata dataset was collected using the iTunes Search API and search terms derived from three dictionary databases [53, 73, 152].

4.2.9 Ethics

Our iOSModZoo and iOSZoo datasets contain publicly-available iOS apps and their metadata. Both app datasets and the random sample of 1 million apps' App Store metadata are available to other researchers (§1.4). Apps were only collected for analysis, not used, except for the case study of the 10 most popular apps and games on the App Store and a few modded versions of each (see §4.3.2). 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.

4.3 The iOS sideloading and modded app ecosystem

This section tackles RQi1: "What is the ecosystem of modded iOS markets and apps composed of, and what is its size?". We leverage insights from our manual analysis of the markets and the static

analysis of the 40 389 app iOSModZoo dataset. We compare the iOSModZoo apps with the 77 189 iOSZoo apps and the random sample of 1 million App Store app metadata.

4.3.1 Static analysis results

We present an overview of the modded app markets and our dataset in Table 4.2, including the apps from each market in our iOSModZoo dataset, the markets' estimated size (based on metadata pages), the unique apps (bundle IDs) and duplicate apps, and the percentage of paid apps in each market. The duplicate apps for each market are identical apps advertised as different apps within the same market.

The markets have an average estimated size of 10 350 apps. The apps downloaded and unique packages columns show many markets in our dataset contain multiple versions of each app, especially iOSGods (5.7 versions per app on average), AppDB (2.6), IPAOMTK (5.8), TopStore (3.4) and Zeus (3.5). These could be occurring all at once, or updated over time during our study like in the case of iOSGods and TopStore where the estimated size is similar to the unique packages and lower than the number of apps downloaded. We also found 502 exact duplicates (exact byte-for-byte copies with the same hash) in more than one market. This is significant since exact duplicates require the signing certificate to be the same as well.

All modded markets offer their catalogue of apps for free, and none of them have payment mechanisms. We found apps that are originally paid apps on the App Store in different proportions in all markets (as shown in Table 4.2) except those we could not scrape fully (see §4.2.3). However, some had a higher proportion of paid apps, hinting at fewer modded freemium apps and more piracy. PDALIFE had the highest proportion, with 61.3% of apps being paid App Store apps offered for free, followed by TopStore with 38.0%, AppDB (20.9%), then Zeus (14.8%) and IPAOMTK (9.2%).

Mobilism works as a forum, its estimated size (10 508 for the iOS side of the market) is taken from the number of 'topics' related to iPhone app releases (4 710) plus game releases (6 006) as of February 2025. Mobilism works as a forum and was not scraped due to its reliance on third-party sites for app downloads that require CAPTCHAs (§4.2.2).

Permissions

We found 6 965 exact matches for the 40 389 modded apps. While 1 021 had completely identical `Info.plist` files, 5 944 had slight changes. However, 6 963 apps had not had their permissions changed, compared to two that had them changed: they both had the 'NSFaceIDUsageDescription' permission added and everything else identical. This might have been done to require FaceID authentication before opening the modded app. This is handled securely by the device and does not leak any personal data. In recent iOS versions (iOS 18 and newer), users can lock any app and require password or FaceID authentication. However, this was not the case in previous iOS versions.

We also found 17 303 closest matches for the remaining 33 424 modded apps. By definition, all 17 303 apps had different `Info.plist` files compared to the originals, including 7 978 (46.1%) with

Table 4.2 Overview of iOSModZoo and the modded markets.

Market	Apps Downloaded	Estimated Size	Unique Packages	Duplicates	Paid (%)
Mobilism	0	10 508	0	0	0
Platinmods	35	1 540	33	0	0
PDALIFE	2 762	2 385	1 663	9	61.3
iOSGods	18 647	4 212	3 278	202	4.3
AppCake	165	30 889	82	1	4.1
AppDB	14 296	46 468	5 463	2 675	20.9
IPAOMTK	5 164	3 461	884	59	9.2
Malavida	35	583	14	0	0
TopStore	5 544	1 646	1 610	342	38.0
Zeus	3 263	1 803	937	1 043	14.8

changed permissions. Closest-matched modded apps typically declare fewer permissions than the App Store versions, possibly because they are older versions in all cases.

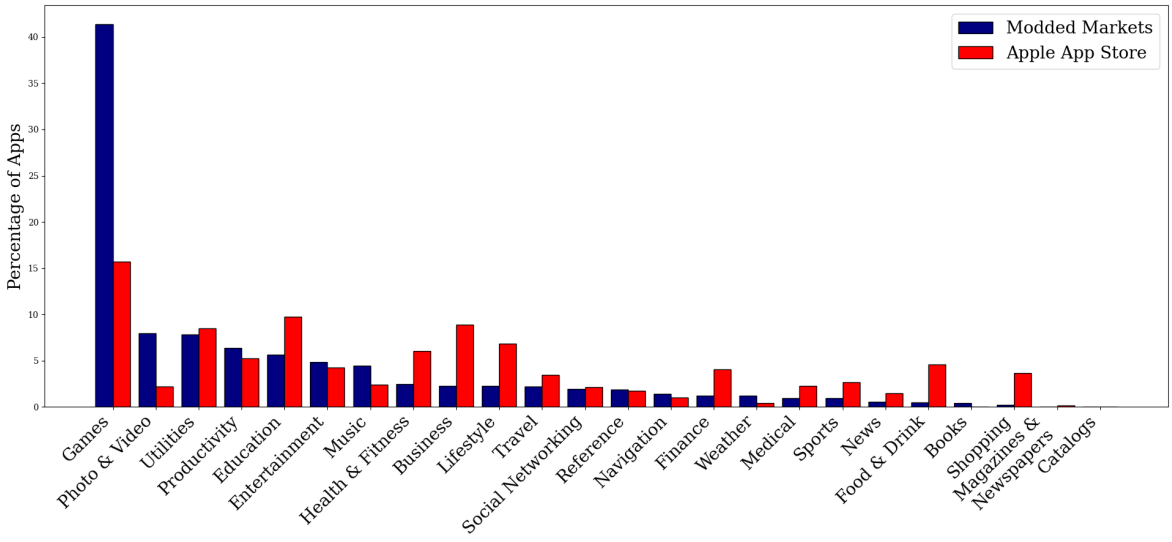
Advertising networks and advertiser IDs

In terms of ad (advertising) networks, of the 6 965 exact matches, 4 285 (61.5%) use Google AdMob, 1 543 (22.2%) use AppLovin, and 1 496 (21.5%) use both. I.e. most of the ones using AppLovin also use Google AdMob. None of them had changed advertising IDs. This is a significant change compared to the Android modded apps studied in Chapter 3.

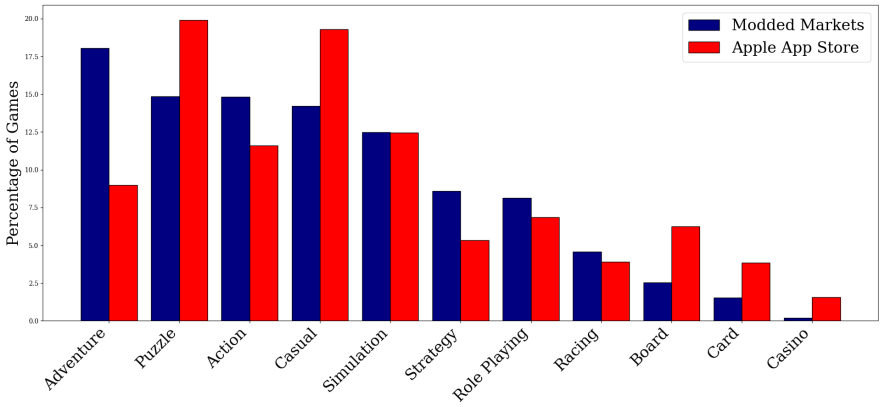
Of the 17 303 closest matches, 5 959 (34.4%) use Google AdMob, 1 666 (9.6%) use AppLovin, and 1 558 (9.0%) use both. Similarly to the exact matches, most apps using AppLovin use Google AdMob too. Many of these apps had added and removed advertising libraries, but none of them had their ad IDs changed compared to the official version. While it is impossible to determine whether the ad libraries were added or removed by the modders, it is possible they were also changed by the official developers between versions. Unlike previous results on Android, we do not observe direct changes to ad IDs except those caused by adding or removing the ad libraries from the apps. Changing the ad IDs requires little effort and is an easy way to reroute advertising revenue away from the original developers to the modders. We find that 120 apps have added Google AdMob, 371 have added AppLovin, 2 649 apps (15.3%) have removed Google AdMob, and 1 440 (8.3%) have removed AppLovin. As a result of these additions and removals 4 825 (27.9%) had at least one ad ID changed, 3 559 (20.6%) had both ad IDs changed, but most (12 478, 72.1%) had no ad IDs changed.

Besides Google AdMob and AppLovin, we also found many other ad networks present in modded and App Store apps. The most common are Aarki, InMobi, Google AdMob, Liftoff, Unity Ads, Appier, Jampp, Remerge and Criteo. Table A.6 in Appendix A.5 shows all ad networks present in 1 000 apps or more (230 out of 472). We found few changes in the overall popularity of the ad networks between modded apps and the App Store.

App categories



(a) Distribution of app categories



(b) Distribution of game subcategories

Fig. 4.1 Distribution of app categories and game subcategories in the App Store and Modded Markets

Games represent an overwhelming majority of the apps in modded markets, which focus on game modifications and pirated versions. Figure 4.1a shows the distribution of App Store categories of the apps found in modded markets and a random sample of 1 million App Store apps. Figure 4.1b shows the distribution of various sub-categories of games that are present in the modded markets. Categories such as Utilities, Photo & Video and Productivity also have a strong presence in the modded markets. Many apps in these categories have functionality locked behind in-app purchases (IAPs) and subscriptions. The modded versions can offer these features for free. Some categories are only present in modded markets: ‘Jailbreak’, ‘Tweaked Games/Apps’, ‘Jailed tools’, ‘Hack Non JB’. These are associated with jailbreaking or content that violates App Store policies, so these are typically not available on the App Store. Some modded markets also include ‘Apple Arcade’ as

a category, to separate games that are originally part of the Apple Arcade subscription (these are typically equivalent to paid games or are versions of free games that have extra content, no ads, etc.).

4.3.2 Case study

Four of the most popular apps and three of the most popular games of all time (as of September 2024) on the App Store are presented as a case study. We manually test official and modded versions to analyse their features and assess the scale of revenue loss caused by modded apps. We tried to achieve a balanced distribution of the markets, although some were down during our study (see §4.2.2).

YouTube

The official YouTube app does not allow background playback including no PIP (Picture-in-Picture) and includes ads. YouTube Premium costs $18.99/month or $29.99/month for the family plan.

Modded versions in AppDB include uYouPlus[4] and YouTubeRebornPlus. Both mods include Premium features: higher resolutions, video and audio downloads, background playback and PIP, no ads, and no popup ads. They also offer extra features: dislike counts using the crowdsourced ReturnYoutubeDislikeAPI, iSponsorBlock for skipping sponsored segments within YouTube videos, etc. Most other markets offer similar versions, although Malavida's is quite outdated (2018).

Spotify

The official Spotify app serves ads to free users between songs, does not allow them to download and listen to music online, limits audio quality to the lowest settings, and forces free users to listen to shuffled albums and playlists. Premium subscriptions remove these restrictions and cost $5.99/month for students to $11.99/month and $19.99 for normal users and families.

SpotUlt++ found in TopStore and Platinmods displays a banner identifying it as 'Spotilife v1.3a' from @ijulioverne and its modded features: unlimited skips, no ads, 'extreme quality sound stream', and playing songs in any order. Higher-quality sound streaming and downloads are enabled server-side. Thus, despite being advertised in some mods, they cannot be included in modded versions.

Instagram

The official Instagram app has no IAPs, paid perks or subscriptions (except for their Meta 'verified' badge which costs $14.99/month). Instagram's revenue comes mainly from ads shown to users.

The 'Instagram Rocket' mod, available in iOSGods and IPAOMTK, allows users to download any picture or video without notifying the author and to view others' stories (temporary photos and videos) anonymously. Users can also repost others' pictures, hide all ads, locally spoof their verified badge and follower counts, etc. IPAOMTK also hosts 'BH', which displays an IPAOMTK logo in the entire app that links to their social networks and website. Similarly to Rocket, this mod hides

[4]https://github.com/qnblackcat/uYouPlus

ads but also suggested posts and suggested accounts. It also allows users to screenshot pictures sent in private messages (even disappearing pictures) without the sender knowing it. The 'MiX' mod allows downloading content, but viewing things privately, hiding ads and suggested posts, etc. are paid mod features priced $1.99/month or $12/year or $4.99/month and $30/year bundled with other mods. IPAOMTK also hosts 'Instagram++', listed as having 'BH, Rocket, MiX, Unlocked All'.

WhatsApp

The official WhatsApp app does not provide IAPs or premium features. Users can adjust privacy settings in ways that minimise effects on others, e.g. users disabling their 'last-seen' or 'online' status cannot see the same information from other users whether they have disabled it or not.

The 'WhatsApp++' mod available in Malavida is outdated. AppDB hosts multiple versions, one has 'Watusi 3', 'Stalky', location spoofing, and 'WAIgnoreCalls'. It includes a small banner with other mods from the same developer, usability and privacy features: users can freeze their last seen status, disable typing indicators, read messages, see pictures and even open multiple times and screenshot 'view only once' pictures without notifying other users. 'Stalky' users get notified when particular contact(s) update their profile picture or status, can change their location including 'live locations' to anywhere they want. These features can enable stalking, phishing, extortion scams and abuse.

8 Ball Pool

The official 8 Ball Pool game offers in-game currencies for $1.99–99.99, premium membership for $7.99/week, and packs, 'lucky spins' and similar for $1.99, $4.99 etc.

Modded versions of the game from AppDB and iOSGods are outdated. The one in IPAOMTK has to be opened in airplane mode and is unplayable due to modded users missing credits that legitimate players are given at the start. Mod features like better aim and strength are not present, and even if they were they would not affect legitimate users, since mod users cannot play online. It is clear that the original developer wants to ensure a great experience for those using the official app.

Candy Crush Saga

Players are limited to 5 lives at a time followed by long waits in the official version. Boosts and perks are needed to pass later levels according to user reviews. The game sells extra moves, lives and 'Gold bars' for $0.99–99.99.

Although Platinmods' and iOSGods' mods are no longer available, IPAOMTK's version is credited to iOSGods. This mod includes infinite moves, lives and boosters, and the ability to turn these on and off. It also includes 'iGameGod',[5] a memory/ cheat engine with capabilities such as finding variables and changing them, slowing down or speeding up games, reducing wait times, and even automating

[5]https://igamegod.app/

Table 4.3 Overview of signing certificates in modded apps.

Market	Analysed	Not signed	Signed	Unique certificates
Platinmods	35	10	25	4
PDALIFE	2 762	2 028	692	31
iOSGods	18 647	10	18 613	19
AppCake	165	120	45	26
AppDB	14 296	12 145	1 880	202
IPAOMTK	5 164	1 170	3 929	202
Malavida	35	11	10	10
TopStore	5 544	0	5 542	18
Zeus	3 263	1 998	1 200	131
Total	48 911	17 256	31 655	275

taps to avoid repetitive parts of games. A mod offered by Zeus is outdated, and AppDB's version is not modded.

Sniper 3D

The official game offers IAPs including: Premium for $9.99/week, $19.99/month, or $99.99/year. Packs with weapons and currency cost $4.99–59.99, coins and diamonds cost $2.99–99.99, etc.

One mod hosted on iOSGods is outdated and only includes 'iGameGod'. Another requires iOSGods login and contains unclosable full-screen ads with redirections. It includes 'iGameGod', infinite coins and ammo, one-hit kills, no reload, infinite energy, etc. Activating more than three at a time requires an iOSGods VIP subscription.

4.3.3 Code signing certificates

As mentioned in Section 4.2.6, we obtain the code signing certificates from the modded apps. We find that 31 655 of the 40 389 apps in iOSModZoo (78.4%) are signed. The rest would have to be signed by users during installation (§2.3.2). Table 4.3 summarises our findings for each market. PDALIFE and AppDB are examples of mainly unsigned app markets, while iOSGods and TopStore almost exclusively contain signed apps, and other markets present a mix of signed and unsigned apps.

The most common certificates and in how many apps they are found in total and in each market are included in Appendix A.6. Some certificates are almost exclusive to some markets in our dataset; iOSGods, IPAOMTK and TopStore have multiple examples of this. The breath of the 'companies' whose certificates are used is vast: iOSGods' main signers are insurance, banking, construction and other companies, while some of the less common certificates are named after individual developers (see Table A.7). It is unclear whether they obtained signing certificates from these companies through impersonation, from employees, or other means.

4.3.4 Market growth

Most modded app markets show a small but steady increase in size when looking at the number of metadata pages during our study. This growth can be seen in Figure 4.2. There are two notable outliers: we observed a notable drop in AppDB in snapshot 2 and AppCake in snapshot 3, suggesting possible maintenance periods or data purges during the data collection phase. We did not include these outliers in Figs. 4.2a and 4.2b, respectively. AppCake's size varied very little during our study, as shown in Fig. 4.2b and its enlarged scale compared to the other markets. In the last few snapshots, many modded markets underwent significant platform changes. Mobilism and iOSGods began implementing Cloudflare challenges, which prevented us from scraping their data. All public download links within AppDB for apps hosted on third-party platforms became hidden after AppDB introduced moderation for user-submitted content that lacked proper metadata and posed potential copyright violations.[6] For TopStore, we were only able to collect metadata for the 1st and 7th snapshots, as the market was inactive during the other periods of data collection.

RQi1: What is the ecosystem of modded iOS markets and apps composed of, and what is its size?

We found 89 modded iOS app markets and obtained 40 389 modded apps from the 9 most popular markets. Most markets have thousands of apps in their catalogues. We found few changes to declared permissions, ad networks and ad IDs (§4.3.1). We tested many apps manually and found subscription features and in-app and in-game items included for free, making IAPs unnecessary (§4.3.2). Most modded apps (78.4%) are signed, many with fraudulent enterprise certificates. All markets show steady growth during our study (§4.3.4).

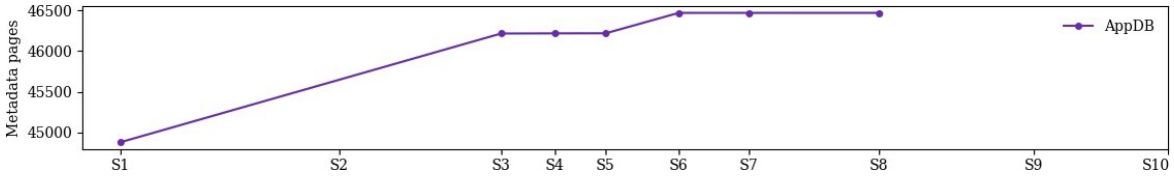
4.4 Market operator motivations and income

In order to answer RQi2: "What are the financial incentives for operating modded iOS app markets? How does this affect the original developers and markets?", we look at markets' monetisation strategies, app metadata including app ratings and number of installs, and the number of users in each market where available. To determine how original developers are affected by these markets, we look at the presence of paid (pirated) apps offered for free in these markets, apps with in-app purchases present in modded markets and common advertised modded features in the markets.

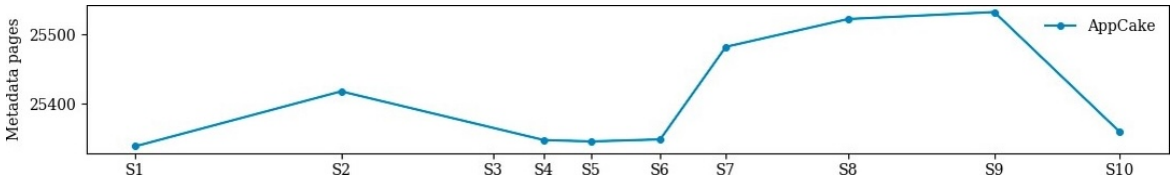
4.4.1 Market monetisation strategies

Unlike Android modded markets (see Chapter 3), many iOS modded markets have paid subscriptions as discussed below. In addition to those below, we had to leave TutuApp out of the study since its free tier was not available for the duration of the study despite the market's popularity. Users wanting to

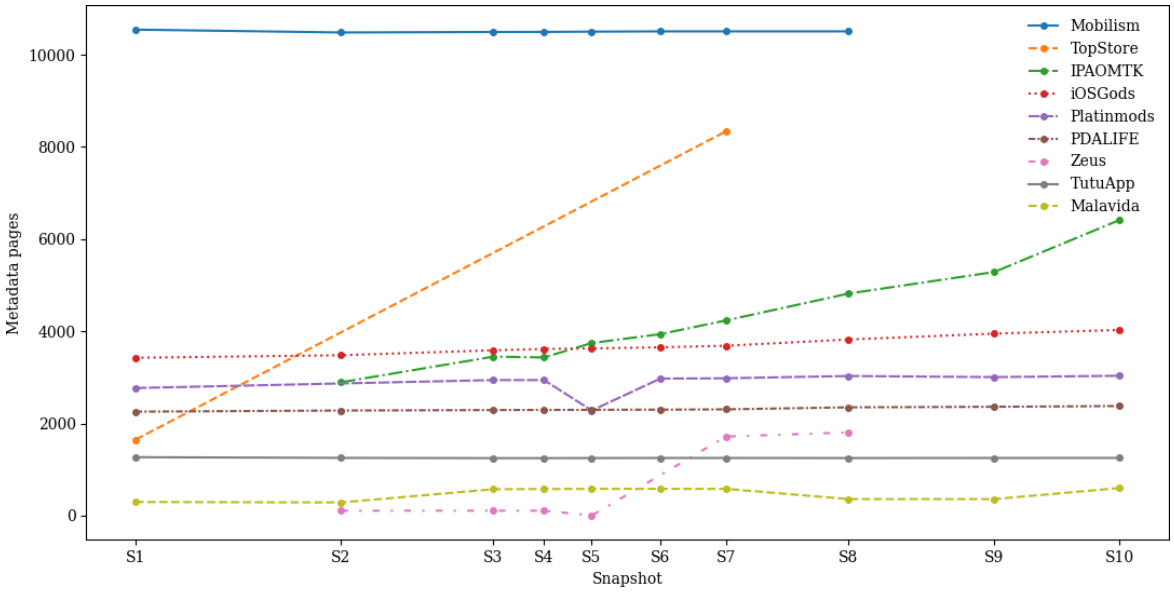
[6] https://appdb.to/news/537



(a) Enlarged view of AppDB size evolution



(b) Very enlarged view of AppCake size evolution



(c) Overview of all other markets' size evolution

Fig. 4.2 Evolution of number of metadata pages in each modded iOS market across different snapshots.

download apps from it had to subscribe to their VIP tier for $17.99/year or forever for $31.99 per device. Furthermore, many iOS markets offer apps via download links from third-party hosting sites, unlike Android markets that typically host apps on their own servers. This makes iOS markets less liable to copyright claims, and reduces their operation costs. The markets operate as follows:

- **Mobilism** works as a free forum with user reputation, comments, upvoting, etc.
- **Platinmods** operates as a free forum with upvoting, different reputation tiers, etc. and it only has a VIP section for Android mods.
- **PDALIFE** offers a premium tier for $10/ three months or $30/year that disables ads on the site and when downloading apps and includes 'exclusive app releases'.
- **iOSGods** operates as a forum with ads. Their premium tier for $34.99/ three months offers unlimited downloads and access to a VIP section of their forum, where they claim apps do not have in-app ads and the forum is free of ads.
- **AppCake** has a market app that is revoked (as of July 2025) and a website. It does not seem to have a paid tier, but it does have a forum section in their website for users to share modded apps on top of the market's main listings.
- **AppDB** directed users to third-party hosting sites to download their apps when we started our study. They later allowed multiple installation methods, including sharing of developer certificates, but this has been changed once again to the following tiers:
  1. The free tier with ads, no support, and limited to 10 installations per week.
  2. 'AppDB PLUS' for €3.99/month or €24.99/year includes unlimited installations, support and no ads.
  3. 'AppDB PREMIUM' for €9.99/month or €59.99/year includes all of the above plus faster dedicated servers and no configuration required for installs (this might mean it is based on enterprise certificates instead of free and paid developer certificates).
- **IPAOMTK** includes ads in its website and offers no premium subscriptions.
- **Malavida** hosts the apps themselves, and does not offer premium subscriptions either.
- **TopStore** has a VIP tier for $29.99/year that unlocks 'exclusive apps and premium features', without clarifying what these are. We assume the VIP apps come pre-signed.
- **Zeus** only includes a donation page for users wanting to support the market.

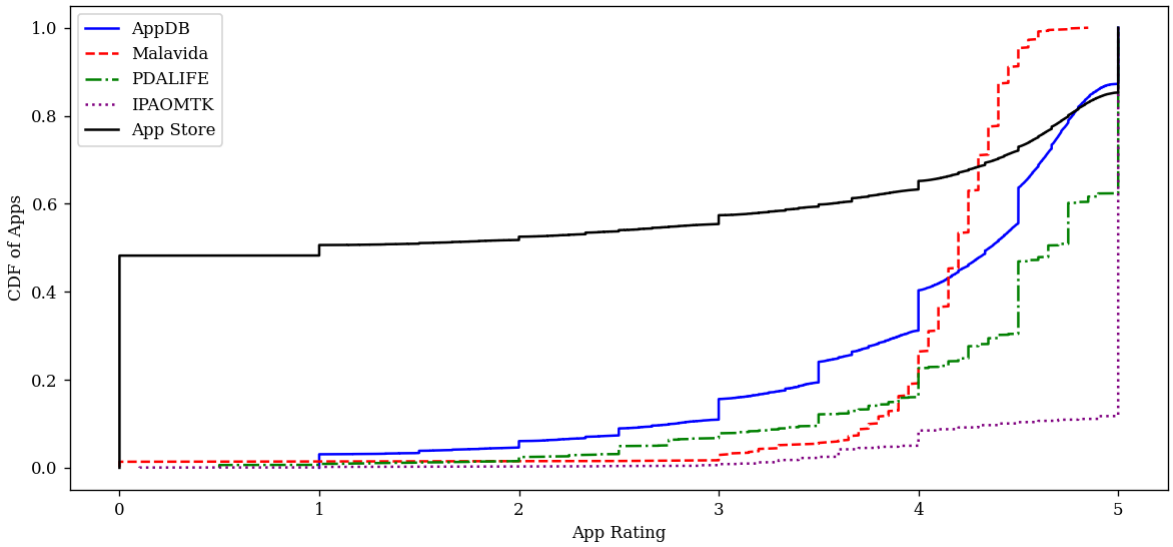


Fig. 4.3 Distribution of app ratings in AppDB, Malavida, PDALIFE, IPAOMTK, and the App Store

4.4.2 Market metadata analysis

Analysing the market metadata in iOSModZoo (89 538 metadata pages) and comparing them with the equivalent App Store apps and the random sample we can gain insights into the way modded apps are presented to and perceived by users and the potential effects they have on the original developers.

Modded features

Apps in modded markets are associated with keywords that suggest unauthorised access to premium features, free purchases and unlimited virtual currencies. Keywords like ‘unlocked’, ‘free purchase’, ‘premium unlocked’ and ‘unlimited money’ appear in a significant percentage of app listings (14.3%, 6.2%, 5.8% and 3.2%, respectively). Other keywords that appear in lower percentages, include ‘no-ads’, ‘Apple Arcade’, ‘jailbreak’, ‘paid for free’, ‘freeze currencies’, etc. These are concerning as they are linked to piracy and unauthorised mods that can negatively impact original developers’ revenue streams from app and in-app purchases, and Apple Arcade commissions and server costs.

App ratings

Apps in modded markets typically have higher ratings compared to apps present in the App Store, as shown in Figure 4.3. Only four modded app markets support ratings. Only 28 679 modded apps (32.0% of the 89.5k metadata pages in iOSModZoo) have ratings, compared to 100% of the 1 022 886 apps from the App Store random sample. In three of the four modded markets, over 80% of rated apps score higher than 4 out of 5. By contrast, reviews in the official App Store tend to be more critical.

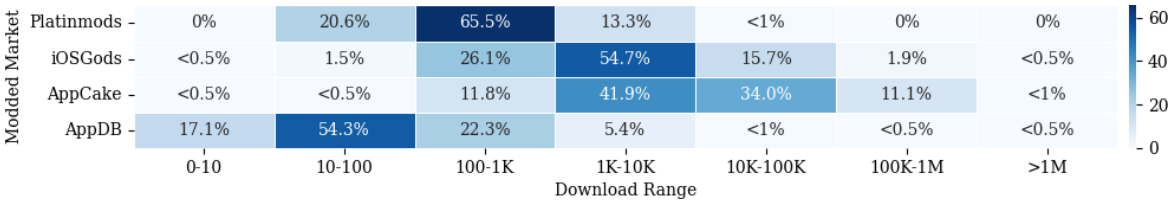


Fig. 4.4 Distribution of app downloads per modded market

App installs

Five iOS modded app markets include the number of installs or downloads for each app in the metadata pages, unlike modded Android app markets. Figure 4.4 shows the distribution of app installs in the Platinmods, iOSGods, AppCake and AppDB markets. The App Store does not publish this information, unlike Google Play.

Platinmods has a majority of apps (65.5%) with 100–1k installs. It is a forum-style market that does not provide the exact number of app installs but only allows users to download each app after liking or commenting on the relevant forum post. We considered the number of installs based on the unique number of users who liked and/or commented on each app post.

The distribution of installs on iOSGods and AppCake is similar. In iOSGods 54.7% of apps have 1k–10k installs and in AppCake 76.0% have between 1k–10k and 10k–100k installs. AppDB has a larger share of low-download apps compared to iOSGods and AppCake. This is perhaps because it does not allow direct downloads from the market without users first linking a device to their market account, and it does not allow more than 10 app installations per week.[7]

Extremely high number of downloads are rare across these markets—none of the markets have a substantial number of apps with over 1 million installs. However, most markets host multiple versions of the same app, and post updated apps as new apps.

Paid (pirated) apps

A big proportion of modded apps (28.9%) are paid App Store apps offered for free in modded markets, as shown in Table 4.4. Pirated apps represent 2 233 unique bundle IDs out of the 7 728 bundle IDs found in iOSModZoo that are available in the App Store. While the majority of modded apps are still free App Store apps (71.1%), this 28.9% of pirated apps is a significant difference from the Android modded markets, where pirated paid apps made up less than 5% of the market (see §3.3.1). The higher proportion of paid apps in the iOS App Store allows modded app markets to simply upload pirated apps and games for free instead of having to modify the code of free apps themselves to include premium features.

To obtain an upper-bound of the estimated revenue loss caused by modded markets, we multiply the original price of each paid app by the number of times it was downloaded from iOSGods, AppCake

[7]https://appdb.to/my/buy

and AppDB. The estimated revenue loss (calculated as US price × number of installs from each modded market) is $12.51 billion. This is obviously an upper-bound and many of these downloads would not have translated as purchases. Furthermore, some installs are possibly users updating to a newer version or trying multiple modded versions of one app. However, this shows the significant size of the problem. This number increases by only a few thousand if we include Platinmods. While the usual piracy debate of how many pirated downloads actually represent lost sales also applies to these modded paid apps (see §3.4.4), we believe these numbers give an indication of the size of the problem. For example, if only 0.1% of pirated downloads would have been a legitimate purchase otherwise, this results in an estimated $12.5 million revenue loss. Previous studies and estimates have suggested high user displacement of up to -2.49 for games, suggesting each pirated download displaces more than two genuine purchases (see §3.4.4). Although we do not know the exact number and displacement, this upper bound shows piracy in modded markets represents a substantial revenue loss for original app developers, and also for Apple since they take a commission of each transaction in the App Store.

Most paid apps originally have a price in the $0.29–$5 range in the App Store. Only about 2.3% of the pirated paid apps are originally priced above $10. We also found 579 Apple Arcade apps in modded markets, corresponding to 221 unique bundle IDs. This means most if not all the 'over 200 games' advertised by Apple as part of the Apple Arcade subscription are available in modded markets for free. Apple Arcade users typically pay $6.99 per month for access to more than 200 games.

Table 4.4 Upper-bound estimation of the revenue loss if all pirated downloads in modded markets displaced a legitimate App Store purchase.

Market	# Paid Apps	Total Downloads	Revenue Loss (in USD)
Platinmods	9	4 223	$11 533
iOSGods	107	4 788 382	$31 513 214
AppCake	25 486	1 102 464 254	$12 382 900 248
AppDB	26 252	8 967 901	$92 520 257
Total	51 854	1.12 billion	$12.51 billion

The download numbers for AppCake in Table 4.4 require further study. Thus, we compiled the download numbers during our study and version number of all available apps and games from our case study: YouTube, Instagram, WhatsApp, Candy Crush Saga, 8-Ball Pool, and Subway Surfers. A random sample of eight of the most downloaded paid apps are also included: only 66 of the 25k paid apps in AppCake have over 100 000 downloads like the popular apps in our case study. The evolution of the number of downloads and version numbers is shown in Figure 4.5, solid lines show the case study apps, and dotted lines the paid apps. Using linear regression, we can estimate that at this rate the apps in Fig. 4.5 would have been added to the market between late 2016 and 2021 (10 of the 14 between 2019 and 2021). Thus, the numbers seem plausible and not fabricated. In reality, installs are probably not linear, and these modded versions could be more recent than we estimate. These apps probably replaced older versions. AppCake opened in 2008.

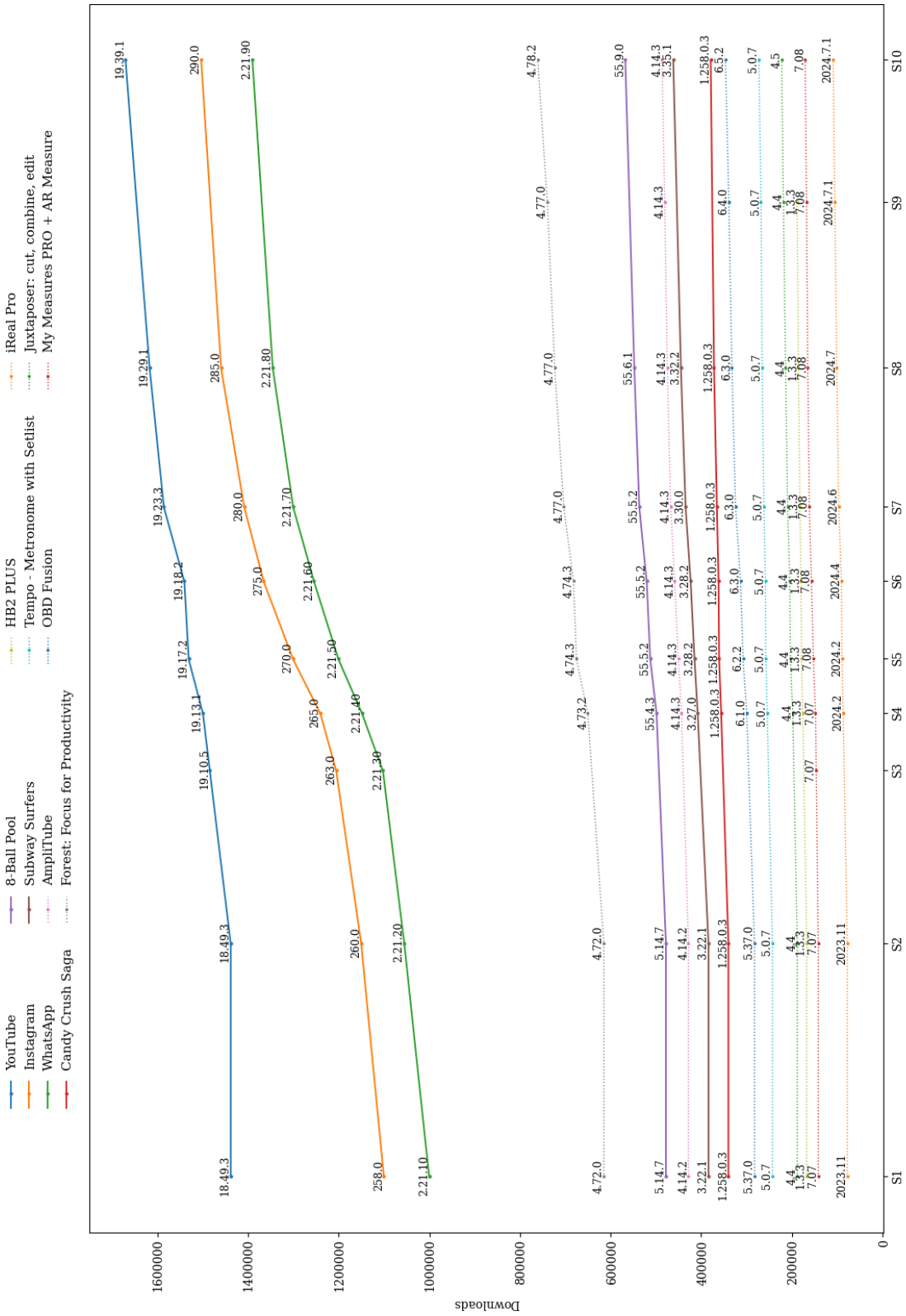


Fig. 4.5 Evolution of (free) case study and (paid) pirated app downloads in AppCake during our study.

In-App purchases (IAPs)

Of the 7 728 unique modded apps (bundle IDs) in iOSModZoo that have an App Store equivalent, 69.0% (5 335) offer in-app purchases (IAPs) in their App Store version. The total original value of these IAPs is $582 142; note the App Store only shows the information for the top 10 most popular IAPs. The distribution of prices is shown in Table 4.5.

Many modded apps grant access to premium features and extra in-app and in-game content through changes to the code. These usually overlap with subscription features and IAPs (see case study §4.3.2). Thus, this translates into potential revenue losses for the original developers and Apple.

Table 4.5 App price and in-app purchase (IAPs) price distribution

Paid apps		**In-app purchases**	
Price Range	**Count (%)**	**IAP Range**	**Count (%)**
Free	5 331 (71.1%)	$0.29–$9.99	663 (8.6%)
$0.29–$4.99	1 610 (21.5%)	$10–$49.99	1 299 (16.8%)
$5–$9.99	380 (5.1%)	$50–$99.99	1 555 (20.1%)
$10–$19.99	112 (1.5%)	$100–$499.99	1 702 (22.0%)
$20–$49.99	57 (0.8%)	$500+	106 (1.4%)
Total	$12 507 USD	**Total**	$582 142 USD

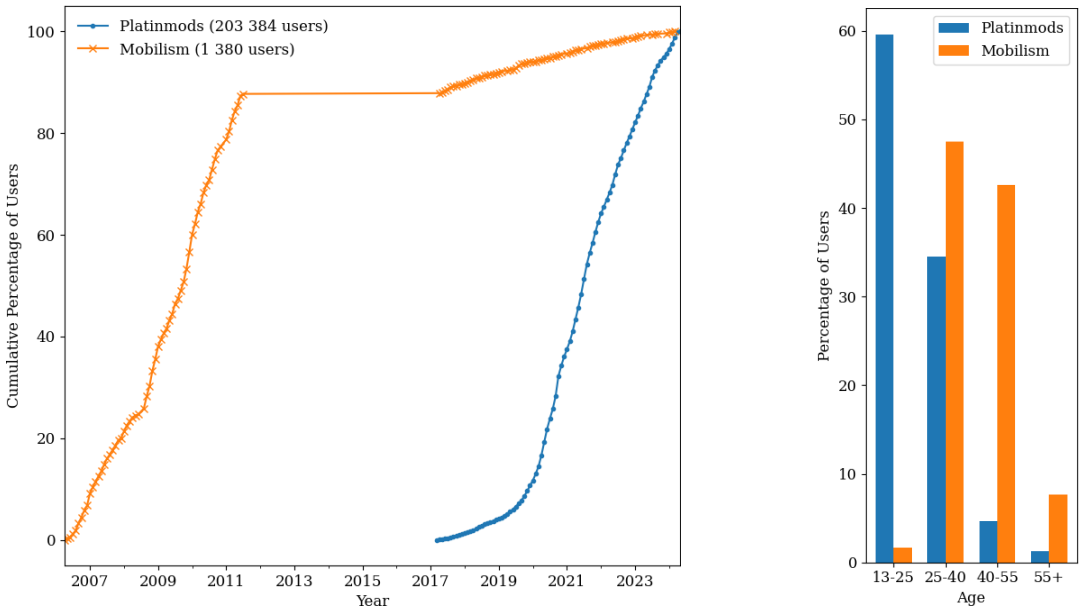
4.4.3 Modded market users

All of these pirated apps and apps with modded features offered to users are only significant with a large user base. The user base of modded markets is not only larger than previously thought, but also very steady. We study the market growth over time and user demographics for Platinmods and Mobilism, the only two markets with user information: Platinmods has 203 384 iOS users and Mobilism only 1 380. Both are forum-based platforms and also include posts related to Android apps and free ebooks. For our study, we focused exclusively on users who posted or commented on threads related to iOS apps in each market. Figure 4.6 shows the cumulative percentage of users over time and their age distribution. Figure 4.7 shows the number of monthly active users (MAU) over time in both markets, this is the unique users that interact with the markets within a 30-day period.

Most of Mobilism's iOS forum users (87.7%) joined the platform before 2012 (Fig. 4.6a). Mobilism users have different ranks depending on their contribution level, 96 users had the ranks: 'VIP', 'Platinum', 'Major Android/iOS/Games Releaser', etc. Interestingly, there are no iOS threads in Mobilism posted by users who joined the platform between 2011 and 2017. This might signal temporary unavailability of the market or shift in platform activity, or the removal of archival content due to copyright infringement or other reasons. Platinmods, in contrast, showed rapid growth after 2019 and has significantly more iOS users compared to Mobilism (Fig. 4.6a). Platinmods also assigns various ranks to its users such as 'Rookie', 'Platinian', 'PMT Elite Modder', etc. The age distribution of these markets' users is shown in Figure 4.6b: Platinmods attracts a much younger audience mostly

in the 13–25 age range. Mobilism, on the other hand, has a user demographic concentrated primarily in the 25–55 age range.

In terms of monthly active users, as shown in Figure 4.7, Platinmods had more than 5 000 active monthly users in some months in 2021, and had about two to three thousand active users in the last few months, as shown in Fig. 4.7. This is mainly because Platinmods requires users to sign in and like or comment on each post from which they want to download a modded app. Mobilism has tens of active users at most in the last years because signing in, commenting, and liking posts is optional in their market. These two markets show a steady and active modded app user community.



(a) Cumulative user growth

(b) Cumulative user age distribution

Fig. 4.6 Cumulative user growth and age distribution in Platinmods (blue) and Mobilism (orange).

Although only Platinmods and Mobilism publish user data, AppDB reported having more than 270 million installations overall [15]. This included 1 180 000 unique installations in the previous 9 months: more than 1 million user-provided apps (this is where most pirated and modded apps are included), followed by apps from third-party repositories and a handful of ‘official apps’ submitted by AppDB themselves and 7 other developers.

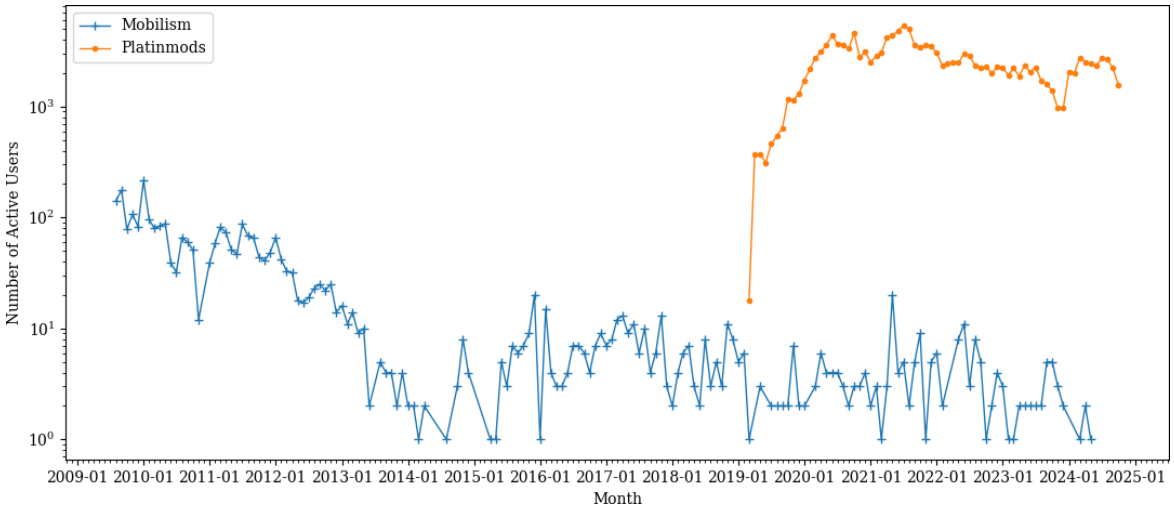


Fig. 4.7 Overview of the evolution of monthly active users in Mobilism and Platinmods.

RQi2: What are the financial incentives for operating modded iOS app markets? How does this affect the original developers and markets?

Many iOS modded app markets have subscription tiers for their users to get exclusive apps and/or quicker updates when app signatures get revoked by Apple (§4.4.1). The markets also include ads in their websites, and many use third-party file hosting sites to reduce their operational costs and liability to copyright claims. They entice users with desirable features like no ads, subscriptions, etc. that potentially erodes the original developers' revenue (§4.4.2). 29% of the modded apps are pirated paid apps offered for free in these markets, their 1.12 billion downloads in just three markets are worth $12.5 billion at the official prices (although it is impossible to know how many would have bought the app otherwise §4.4.2). We also found 221 unique Apple Arcade games, representing most of the subcription's catalogue. Almost 7 in 10 modded apps originally have IAPs in the App Store. These numbers are significant because these markets are not going away (§4.4.3).

4.5 Security implications of modded apps and markets

This section tackles RQi3: "What are the security implications of installing apps from these iOS markets?". We revisit previous results and look at VirusTotal analysis results in Section 4.5.1. The security implications of trusting third-party signing certificates, mobile provisioning profiles, and the more invasive Mobile Device Management profiles are explored in Section 4.5.2.

4.5.1 VirusTotal analysis results

As explained in Section 4.2.7, we obtained limited existing results from VirusTotal, requiring us to send the apps to be analysed for the first time. This in itself is a security concern, as it means most modded apps are never analysed by users before being installed nor by markets before being hosted (at least not with the around 60 antivirus engines used by VirusTotal). We obtained results for all 40 389 modded (iOSModZoo) and 77 189 App Store (iOSZoo) apps. With a threshold of at least 3 antivirus engines marking each malicious app, we find:

- 22 malicious apps in iOSZoo (App Store apps dataset) (0.03%), summarised below.
- 406 malicious modded apps in iOSModZoo (1.01%), summarised in Table 4.6.

All labels obtained from VirusTotal for the 22 apps marked as malicious in iOSZoo are the following: 149 Exploit/ Exploit-kit including 140 Exploit.CVE-2011-3402, 120 Malicious, 26 Malware.Trojan. The labels obtained from VirusTotal for the malicious iOSModZoo apps are summarised in Table 4.6. Exploits are the predominant label in apps from both datasets. The second most common label for modded apps is Malware. Adware was not detected in the App Store apps, and was not that common in modded apps compared to exploits and malware.

4.5.2 Trusting signing certificates and Mobile Device Management (MDM) profiles

Trusting unknown signing certificates is not inherently more dangerous than self-signing when installing potentially dangerous sideloaded apps. However, as explained in Section 2.3.2, many markets use Mobile Device Management profiles which are meant for organisations and companies to control company devices. This gives the profile owner the ability to remotely install apps to the devices (which the markets promise to do only with user consent), change configurations, log users out, querying the list of installed apps, serial number, Unique Device Identifier, MAC address, etc. Thus, users put a great amount of trust in the market operators not to abuse these features, since modded apps are usually installed on users' personal devices.

4.5.3 Lack of analysis tools without Jailbreak

There are no third-party malware analysis tools in iOS that can scan and analyse other apps and their files due to app sandboxing. This strictness of the set of permissions apps can obtain in iOS is a useful security feature in a walled garden ecosystem. However, once sideloading is possible, it becomes a security concern that iOS lacks client-side scanning of installed apps equivalent to Google Play Protect on Android.

For markets that allow users to download the IPA files and sign them themselves, users could scan the files from their phones or computers using online tools like VirusTotal before installation, but our results show most of them do not (§4.5.1). Furthermore, most users without a paid developer account or an old device need to download pre-signed apps or use the MDM profiles offered by most markets.

Table 4.6 All iOSModZoo malicious apps labels.

Label	Count
Exploit/ Exploit-kit	328 (Total 4 400)
Exploit.Jailbreak	1 346
Exploit.CVE-2023-32434	14
Exploit.CVE-2022-46689	1 255
Exploit.CVE-2021-30937	80
Exploit.CVE-2019-6225	127
Exploit.CVE-2018-4243	7
Exploit.CVE-2016-7644	8
Exploit.DirtyCow (CVE-2016-5195)	597
Exploit.CVE-2011-3402	65
Exploit.Vortex	230
Exploit.AsyncWake	150
Exploit.Kfd	94
Exploit.WeightBufs	43
Exploit.Kappotoma	36
Exploit.Dakkatoni	11
Exploit.Trident	7
Exploit.opainject	2
Malware	123 (Total 1 807)
Malware.Trojan	1 568 (Total 1 588)
Malware.Trojan.MiniUPnP	13
Malware.Trojan.Skeeyah	4
Malware.Trojan.Multiverze	3
Malware.Siggen	75
Malware.Spyware	13
Malware.KeyRaider	5
Malware.PALLASNET	3
Malicious	1 324
Suspicious	122
Riskware	91 (Total 202)
Riskware.Miner/.BitcoinMiner	111
Adware	57 (Total 64)
Adware.GuiInject	7
PUP/PUA	13
Revoked Certificate	4
Backdoor	1

Besides the concerns described in §4.5.2, MDM profiles result in direct installations of the apps when the user clicks 'install' on the modded market app or website. Most markets using MDM profiles even allow users to install apps remotely from a different device, e.g. click 'install' from their computer to have the apps remotely installed on their registered iPhone or iPad. However, this means users do not have access to the IPA file before nor after the installation, which prevents them from being able to analyse the files for malware with tools like VirusTotal.

Furthermore, analysing already installed apps requires a jailbroken iPhone, which creates difficulties for users and researchers. Jailbreaking an iPhone in 2025 requires a very outdated version of iOS (iOS 16.5 or older, released May 2023), and an outdated phone (iPhone 14 Pro from 2022 or older). This generates security concerns in itself, as those iOS versions are missing two years of security updates. Even though iOS 16, 15, and older still receive security updates, the versions needed for jailbreaking are usually outdated as they need to exploit security bugs during installation, e.g. iOS 16.5 instead of the latest iOS 16 version, 16.7.11.

4.5.4 Countermeasures, market changes and third-party hosting

Modded markets implement various countermeasures to prevent scraping and automated app downloads. Unlike modded Android markets that host modded apps on their own servers (see Chapter 3), most iOS markets host apps on third-party file hosting platforms. One market used torrent files. Hosting platforms require users to wait before making successive app downloads, implement Cloudflare protection, and use CAPTCHA mechanisms to prevent automated downloads. This exposes users to more ads and phishing sites. Furthermore, these third-party hosting site and torrent links can point to different files over time, so a user can post a link to a genuine modded app download and get upvoted (good reviews) in forum-like markets and later change the linked resource, doing all kinds of damage. Previous research found that a better reputation in criminal pseudonymous forums can lead to higher success for criminals [74]. Similarly, users (and market operators) might not test the security of apps they download that were posted by highly reputed pseudonymous users. One market requires users to comment on or 'Like' each post to make download links visible. We reverse engineer a couple market apps to obtain direct download links to collect all modded apps. Another had CAPTCHA protection mechanisms for app downloads, so we manually collected the apps marked as 'Tweaked'. To avoid legal and ethical concerns, we did not download any torrent files.

Table 4.7 Comparison between official and modded Android and iOS market number, size, and pirated and paid apps.

	Markets found	Average market size	Maximum market size	Paid and (pirated) apps
Google Play	N/A	3.55 million [208]	N/A	3.1% [208] (N/A)
Android modded markets	414	37 486	221 089	0% (4.7%)
App Store	N/A	1.64 million [208]	N/A	5.8% [208] (N/A)
iOS modded markets	89	10 350	46 468	0% (28.9%)

RQi3: What are the security implications of installing apps from these iOS markets?

Modded iOS apps are not usually analysed by users before being executed, unlike Android sideloaded apps that are analysed by Google Play Protect (and regular apps are analysed before being added to Google Play and the App Store). Most had never been seen by VirusTotal; 1% of modded apps were classified as malicious (33 times more) versus less than 0.03% App Store apps (§4.5.1). Modded apps were mainly classified as having CVE and other exploits, and trojans. Trusting unknown signing certificates and especially mobile device management (MDM) profiles entails many risks for users, including trusting markets not to abuse MDM profiles to extract private data, remotely install apps without user consent, etc. (§4.5.2). We briefly explored the difficulties for researchers, users, and apps when trying to analyse apps on modern iPhones (§4.5.3). Many markets rely on third-party hosting sites for file downloads, exposing users to more ads and phishing sites; third-party links and torrent files are out of the control of market operators and users and can point to different files after some time (§4.5.4).

4.6 Comparison of iOS and Android modded apps ecosystems

Table 4.7 shows the differences in terms of number of markets and their size, as well as the number of paid apps in official markets and pirated paid apps in modded markets. Table 4.8 summarises the differences between Android and iOS modded markets in terms of modded permissions and advertising libraries, as well as their differences with the official markets in terms of malicious apps. Further differences between the markets are discussed below. It is worth noting the different scale and duration of the studies (iOS was studied over a longer period but included on fewer markets and apps). Thus, we have used percentages in the tables where appropriate.

Table 4.8 Comparison between official and modded Android and iOS market modified permissions, advertising libraries, and malicious apps.

	Modded permissions	Modded ad libraries	Malicious apps	Currently-available malicious apps
Google Play	N/A	N/A	1.7%	0.9%
Android modded markets	28.8%	11.1%	6.8%	6.8%
App Store	N/A	N/A	0.03%	0.03%
iOS modded markets	0.0%	0.0%	1.0%	1.0%

RQi4: How does the ecosystem of modded iOS apps differ from its Android counterpart?

While there are fewer sideloading markets in iOS than in Android (89 compared to 414), they have billions of downloads and steady user bases. Many offer subscriptions, ads within the market, and use third-party hosting sites to reduce their operational costs. No Android markets offered subscriptions, and they mostly hosted the apps themselves. Android modded apps tended to have changed advertising networks and ad IDs, while this is not the case in iOS.

Less than 5% of modded market apps in Android, compared to 29% of the modded iOS apps in the markets analysed are pirated paid apps offered for free. The iOS apps collectively have over 1 billion downloads worth $12.5 billion in just three markets. All or most of the 'over 200 games' Apple Arcade catalogue is available for free as well.

In terms of security, both iOS and Android modded apps have lower security than their respective counterparts from official markets, unsurprisingly. While Android modded and code-modded apps were 7.6 and 9.6 times more likely to be marked as malicious than available Google Play apps (6.8% and 8.6% versus 0.9%, respectively, see §3.5.1), iOS modded apps are 33 times more likely to be classified as malicious by VirusTotal than their App Store counterparts (1% versus 0.03%, respectively, see §4.5.1).

Furthermore, there is also a false sense of security in iOS resulting from fewer users being able to analyse the apps they sideload before installation. Most apps had not been seen by VirusTotal. This makes antivirus engines potentially less accurate for all users.

4.7 Limitations

Our modded app scraping was limited, as mentioned in Section 4.2.2 due to: the unavailability of some markets during part or the entirety of our study; the presence of CAPTCHA challenges embedded in the markets, and some markets' use of third-party file hosting sites. To minimise this, we performed manual scraping of the markets containing CAPTCHAs. We also manually obtained session cookies

and fed these to our automated scrapers to circumvent the need to manually scrape some markets (§4.2.3). Additionally, we developed scrapers for the most popular third-party hosting sites.

Ad networks, equivalent to Android ad libraries, are listed in the `Info.plist` file by their IDs (§4.2.6). We tried to obtain a list of the most common ad network IDs from different sources to correlate them with companies like Google AdMob, IronSource, etc. but this is incomplete and Apple does not publish an official list.

Another limitation compared to previous literature is the way apps are signed by Apple (in the App Store), and by users themselves, using developer certificates, enterprise profiles, etc. in modded markets which leads to exact code matches not being exact matches byte-for-byte unlike in Android. This makes quick comparison of code at scale impossible. Furthermore, as mentioned in Section 4.5.3, extracting the unencrypted code from App Store and even some sideloaded apps would require a jailbroken iPhone (thus, an old model) and other tools to extract the apps. As mentioned before, most jailbreaking and app extraction tools are very outdated. The latest jailbreaks are only compatible with versions of iOS that are several years old running in iPhones several generations old. This also affects VirusTotal results, which are unavailable for App Store apps, and very rare for modded apps. Thus, another limitation and security concern has been the lack of previous VirusTotal results (see §4.5.1), which we solved by contributing the full iOSZoo and iOSModZoo datasets to VirusTotal.

4.8 Literature review

The role of modded apps and piracy in displacing users and affecting developers' revenue was already discussed in the previous chapter (§3.4.4).

Prior work has studied different aspects of the App Store and the apps available in it. Deng et al. presented iRiS, a static and dynamic analysis tool they used to analyse two thousand App Store apps [68]. They found a few that abused private APIs. The tool decrypts application binaries using IDA Pro, which costs over $1 000/year.[8] Brumen et al. found that free apps collect more data than paid ones in both Apple's App Store and Android's Google Play [49]. They also found privacy labels shown to users in each store were lacking permissions mentioned in the app's privacy policy text. Scoccia et al. analysed the App Store privacy labels Apple shows users in app download pages, and found free apps collect more sensitive data than paid ones, mostly in deanonymised form [183]. A wide range of sensitive information is collected for tracking purposes. Liu et al. analysed third-party SDKs used in iOS (App Store) apps and found 47% of 32 thousand popular apps analysed used SDKs that did not adhere to privacy laws or Apple's policies [138].

Orikogbo et al. compiled a dataset of 43 thousand iOS (App Store) apps to identify common third-party libraries and SSL certificate analysis [157]. They found average apps in 2016 were mostly library classes plus less than 40% of developer-authored code. Their analysis pipeline is based on Clutch [125], a tool that extracts unencrypted iOS app binaries from jailbroken iPhones as they are

[8] https://hex-rays.com/pricing

unencrypted during runtime. Others like Feichtner et al. also based their analysis tool on Clutch and statically analysed binaries dumped out of jailbroken iPhones to find more than 82% of the 417 apps they analysed violated a security rule [83, 125].

Tiganov et al. introduced an open source static analysis tool for Swift, but it needs to be run on the unencrypted app (on the XCode project before the developer publishes it) [206]. Guo et al. implemented iLibScope, a third-party library detection tool for iOS apps [108]. The decryption is again based on IDA Pro. They identified vulnerable libraries used in real-world apps. Their approach is based on the creation of profiles for each target library and each app analysed, similar to those mentioned in the previous chapter in §3.6 [38, 141, 217, 223]. They work well to find specific vulnerable library versions, but are unfeasible for general library detection, especially in datasets of iOSModZoo's scale. Tang et al. obtained a 168 thousand app dataset from the App Store, similar to the methods already mentioned above, they require a jailbroken iOS device for decryption [195]. They analysed 1 300 apps in detail to create signatures used in the large-scale analysis of their dataset. They found 92 apps that use third-party libraries which listen for remote connections, potentially opening them to data leakage, remote command execution, and denial-of-service attacks.

There is little prior research on sideloaded iOS apps. Wei et al. discussed the targetted attacks that are possible using enterprise provisioning profiles in theory [220]. Zheng et al. statically analysed the `Info.plist` files and performed dynamic analysis on 1 408 apps obtained from sideloading markets [224]. They found 64% send sensitive information through unencrypted channels or store it in plaintext. As mentioned in Chapter 3, prior work had identified harmful libraries in Android and iOS both in the official stores and sideloading markets, although they had a smaller sample of the latter [58].

4.9 Summary

We have found that Apple's policies aimed to make their ecosystem more secure for mobile users have been successful in some ways: sideloading is not as popular as on Android (89 markets compared to 414 Android modded app markets), Apple has added a lot of friction to the process and they have retained control of their platform. Cracking down on enterprise signatures and MDM profiles could reduce the modded apps and sideloading problem.

However, it is still a very popular ecosystem, filled with pirated and modified apps. We have found over a billion downloads of pirated paid apps worth $12.5 billion (in just three markets, although it is unclear how many users would have purchased the apps officially if modded markets did not exist §4.4.2), as well as the almost full catalogue of Apple Arcade games. Furthermore, security-wise it has also caused systemic issues in their platforms, with users that wanted to install apps that are not allowed in the App Store like emulators, clipboard managers, virtual machines, etc. having to resort to jailbreaking or sideloading methods. Some are simply tedious like signing your own apps using a computer, while others require exploiting bugs e.g. TrollStore, or exploiting tools from Apple meant for enterprise testing and provisioning. The latter gives market operators a level of access and control

over the user devices that if abused would be a much bigger privacy and security concern than any sideloaded apps. Furthermore, the modded apps themselves are less secure than their counterparts from the App Store, similar to Android modded apps. Modded iOS apps were tens of times more likely to be classified as malicious by VirusTotal (1.0% versus 0.03% in the App Store). The amount of malicious apps in the App Store likely correspond to noise.

As mentioned above, possible short-term solutions include a more proactive approach to banning and revoking certificates from Apple. Maybe by analysing random samples of apps from these modded markets.[9] Apple already periodically tackles this, and stated in March 2024 they "stopped 3.8 million attempts to install or launch apps distributed illicitly through the Developer Enterprise Program" [30]. A few markets struggled during our study to sign apps for users, while many offered multiple installation methods and managed to obtain more developer and enterprise certificates to continue their operations unfazed.

While apps in third-party app markets in the EU that have opened as a result of the Digital Markets Act (DMA) are still notarised by Apple and security checked, this is not the case for apps signed by Developer Accounts or users themselves, or through MDM profiles (see §4.5). Users are defenceless without any client-side scanning or antivirus scanning possible when apps are installed directly on their devices if they are enticed by modded features, open source apps, emulators or other apps prohibited in the App Store, want to test paid apps for free, etc. Thus, despite the frictions introduced by Apple to prevent sideloading, it could be argued that once users have sideloaded apps, Android protects them better with their Google Play Protect client-side scanning tool that periodically scans user devices for risky software.

Furthermore, it will always be difficult to prevent users from using self-signing sideloading methods such as AltStore or sharing paid developer accounts (see §2.3). However, it is reasonable to argue that self-signing methods are less likely to be used by mistake or by non-tech savvy users. Thus, it is important to proactively prevent the quick, few-click sideloading methods that put users at high risk, and which also affect developer revenue streams.

[9] https://developer.apple.com/support/certificates/

Chapter 5

Surveying modded app market operators and original app developers

This chapter presents the results of the first survey-based study of 14 Android and 5 iOS modded app market operators and the effect of modded apps on 717 original app developers. We find modded market operators have economic incentives to break copyright law and make it difficult to file complaints. They perform little to no security testing of the apps they host.

Previous chapters studied the technical features and metadata of apps found in Android (Chapter 3) and iOS (Chapter 4) modded app markets. In this chapter we survey Android and iOS modded market operators to understand the motivations, economic incentives, security protocols, and general operation of their markets. We also aim to understand the original app developers' awareness, estimated revenue loss, other perceived impacts, and their experiences with the tools available to prevent and combat modding including technical tools and Digital Millennium Copyright Act (DMCA) takedown notices. For this, we leverage the updated ModZoo dataset [176] to obtain contact details of 27 629 original app developers affected by Android modded app markets. We invite them to participate in our survey and obtained detailed responses from 717 developers affected by modded apps. We find modded markets benefit from the original app developers' intellectual property while affected developers incur revenue losses from purchases, advertising, increased server calls, etc. and negative impacts from increased errors, support requests and reputation damage. We also study DMCA compliance of the top 23 Android and iOS modded app markets and confirm our survey findings: DMCA copyright claims are unusable at scale. Finally, based on these findings together with the survey responses, we propose possible solutions to the challenges presented by modded app markets.

In summary, this chapter makes the following contributions:

- The first survey-based study of Android and iOS modded market operators (§5.2): they have economic incentives, mainly monetise their markets through ads, perform little security testing of apps and have misconceptions regarding iOS and Android security.
- The first survey-based study of 717 original developers of all sizes and popularity affected by modding (§5.3): developers are very familiar with Android modded apps, but less so with iOS.

- Developers of all sizes suffer revenue losses; increased costs and support requests; and reputation damage (§5.3).
- A study of DMCA compliance in 23 modded markets, confirming the developers' survey findings that DMCA is not usable (§5.4).
- Propose solutions to the challenges presented by modded app markets for developers, manufacturers, and legislators (§5.5).

5.1 Research questions and methods

Our research involved surveying operators of modded markets (§5.1.1) and the original developers of applications found on the modded markets (§5.1.2). This section details our methodology, including identifying our research participants, our survey instruments, and contacting potential participants. In Section 5.1.4 we outline how we verified DMCA-related issues. We obtained ethical approval for both surveys (§5.1.5). We outline the limitations of this study in Section 5.1.6.

The research questions we seek to answer are the following. RQs1 and RQs2 concern our main research questions RQ1 and RQ2, respectively; RQs3–RQs6 answer RQ3 from the developers' perspective; and finally RQs7 and RQs8 provide the answer to RQ4:

RQs1 What motivates operators to run modded app markets?

RQs2 What are modded market operators' revenue streams?

RQs3 What types of security testing do they perform, if any?

RQs4 Which developers have had their apps modded?

RQs5 Are developers aware of modded app markets?

RQs6 How do modded and pirated versions affect developers?

RQs7 What tools are available and used by developers?

RQs8 Are DMCA notices effective and used by developers?

5.1.1 Modded markets survey

We contacted 414 Android and 81 iOS market operators to understand their motivations, economic incentives, security protocols, and work. We received 14 (Android) and 5 (iOS) responses.

Surveying modded market operators

We use the lists of 414 modded Android and 99 modded iOS markets from the previous chapters. All Android and iOS markets found have an English version, some also in other languages. We manually verify they contain apps marketed as modded.

We sent survey invitations to all 414 Android market operators but only 81 iOS market operators since 18 did not provide contact information. The surveys were delivered using the Qualtrics platform with initial invitations to participate followed by three follow-up invitations over the course of four weeks. Survey participants were sent an invitation link using publicly-available contact information; some market operators had multiple contact details. The 414 Android operators were contacted through 330 email addresses, 32 contact-us or feedback forms, 55 Facebook and 11 Instagram links; the 81 iOS operators through 55 email addresses, 20 contact-us or feedback forms, 27 Twitter, 21 Facebook, 6 Telegram and 5 Instagram links.

The survey instruments are provided in Appendices B.1.1 (Android) and B.1.2 (iOS). The surveys take approximately 30 minutes to complete. They contain 32 open and closed response questions with potential follow-up questions depending on the answers provided. These are divided into five categories: platform-related questions; security checks performed on apps; goals and motivation; operator-related questions; and additional questions. The questions include standard demographic questions followed by questions about the operation of their markets based on knowledge gaps in our previous research from Chapters 3 and 4.

We developed our survey instruments based on best practices, combining multiple-choice questions with open-ended ones for those willing to include more detail in their answers [42, 155, 187]. The survey instruments used for iOS and Android are almost identical except for nomenclature to facilitate direct comparison. The surveys were tested on colleagues to ensure the questions were clear to people with different technical backgrounds and non-native speakers. Questions about monetisation, revenue and operation costs are formulated in terms of percentage of their income coming from and being used to run the markets. This is common practice since we are interested in understanding how their market operates and their motivations [113].

5.1.2 Original app developers survey

We directly surveyed original app developers whose apps have been found in Android modded app markets. We gathered insights on their awareness of these markets as well as the inclusion and modification of their apps. We also asked about their experience with the tools available to developers to report and prevent modding, their estimation of revenue loss and other impacts. We asked about their experience and success (or otherwise) with copyright claims, take-down requests, and their use of the DMCA forms. Finally, we asked them their position at the company, the number of employees, and accepted an open response for further comments. The survey instrument is provided in Appendix C.1.1.

Identifying original app developers

We leveraged our ModZoo dataset, and updated it to include 280 981 modded Android apps from the 13 modded markets in Chapter 3. We then obtained the package names of the modded apps and found the 46 541 equivalent unique Google Play apps, obtained their Google Play metadata,

including the publicly-available developer emails, as well as the number of installs and reviews. Thus, we obtained 27 629 unique developer emails with an average of 1.68 apps each. The lack of iOS developers with modded apps from iOS markets is due to the App Store not including contact details for app developers, which makes it impossible to obtain accurate contact details for iOS developers at scale. However, even with this limitation, more than 54% of the respondents are also iOS developers (see §5.3.1) which is comparable to other public figures [118]. Therefore, we believe our survey results would not be too different had we been able to contact iOS developers directly.

Developer report generation and emails

We ranked the 27 629 developers by number of installs of all their apps combined. We first tested our survey on colleagues with varying levels of experience developing apps. From the responses, we identified and reworded ambiguous survey questions. We improved the wording of our questions and divided the list of developers into thirds of equal size corresponding to lower, middle and higher numbers of installs. We ran our survey on a random sample of 10% (2 780) of developers as a pilot study. We oversampled the middle third (48.2% of sample) compared to the other two (25.9% each) to maximise response numbers, based on our intuition that email addresses of larger companies would be less likely to be monitored by someone able to answer our survey and would thus take longer to answer, if at all. Furthermore, developers with fewer installs may no longer be active, or fail to monitor their developer email full time.

Finally, we surveyed all remaining (90%) developers and added their responses to those from the pilot (10%). We sent each developer an initial invitation to our survey, followed by two reminders over a month. We waited at least a week and a half between emails and sent them during different weekdays to maximise participation.

Our emails included an introduction to the project and modded apps, the typical ways modded apps are changed, the link to our survey and information sheet. To encourage developers to engage with our survey and inform them, our email also included a list of the modded markets each app is present in out of those studied in ModZoo, and the earliest and latest dates that they were present. Finally, the emails included a list of the 13 modded markets included in the study and their URLs in case developers wanted to check the markets themselves for newer versions of their apps, or familiarise themselves with the markets. While more information could have been obtained from the ModZoo dataset and included in these summaries, we found this a good trade-off between conciseness and valuable insights for developers. It seems we achieved this goal, as we received numerous responses thanking us for bringing the situation to their attention.

5.1.3 Response coding

Coding consists in assigning short identifiers to important ideas, beliefs or facts expressed in surveys to obtain summarisable and comparable results from open-ended or multiple-choice responses [42,

Table 5.1 Top level codes and explanations or examples

Revenue loss	including causes and effects perceived
Modifications	as perceived in modded apps by developers if they analysed modded apps
Other impacts of modding	e.g. demoralisation, impact on other users, reputational, increased exposure
Protection techniques	methods used to detect and prevent modding
DMCA-related actions	previous DMCA actions taken and whether they have been successful, future plans
Google-Apple solution	whether Apple and Google or search engines should offer a solution; experience with and effectiveness of current solutions by Google
Beliefs	reiterated beliefs not in previous codes

155, 187]. Multiple researchers or 'coders' can perform the coding to compare results and thereby ensure the approach is robust (see §5.1.3).

We summarised the Android and iOS modded market operators' responses directly. This was possible due to the structure of the surveys and the more limited number of responses. We manually coded the 717 original app developer responses using NVivo, a computer-assisted qualitative data analysis tool. Following best practices [42, 155, 187] we made our top-level codes in our codebook (the hierarchy of all codes and subcodes) the core themes of our survey, as shown in Table 5.1.

We then coded the responses in random order, adding the subcodes as needed inductively. For example, for *Other impacts of modding* most of the subcodes are negative impacts such as *reputational_damage* and *support_requests_modded_apps_or_features_error*. Subcodes of *positive_impacts* include *more_exposure* and *online_ discussion_more_users*. This iterative process led to some anecdotal codes with only a single reference. However, the codebook in Appendix C.2 includes all codes with two or more references.

In some cases, participants reiterated an opinion or observation throughout their survey responses. Most commonly this happened with the *Beliefs* theme. For example, subcodes related to Apple doing better at tackling sideloading or how it is not worth fighting modding were mentioned in many contexts. When counting the references to each code, we only counted each reiterated code once per participant for more meaningful insights into the number of developers affected or who have a particular opinion. Otherwise the results would be skewed by developers repeating the same point on purpose or without realising in multiple open-ended questions.

Inter-coder agreement

I was in charge of creating the codebook as I was the most familiar with the ModZoo dataset, modded apps and markets. Due to the massive scale of our study, the objectivity of the coding was confirmed by checking a random sample of 10% of the 717 responses with an additional coder more experienced with qualitative research (Professor Alice Hutchings), who was given the full codebook and responses.

Our approach follows best practice [42, 155, 187]. The resulting inter-coder agreement was very high at 95–100% between all main codes with an average of 98%.

We calculate the inter-coder agreement between coders A and B based on the sum of their agreement on the binary presence of each code for each respondent in the sample. In other words, for each code c out of C for each respondent n out of $N = 72$ we calculate the XNOR or if only if (iff $\iff$) of the presence of the code for each coder:

$$\frac{\sum_{c=1,\ n=1}^{C,\ N} \neg(A_{c,n} \oplus B_{c,n})}{N \cdot C} = \frac{\sum_{c=1,\ n=1}^{C,\ 72} A_{c,n} \iff B_{c,n}}{72 \cdot C}$$

Where $N = 72$ for this random sample and C is the number of subcodes and subsubcodes etc. in that particular branch, and $A_{c,n}$ and $B_{c,n}$ represent coder A and B's presence of code c for respondent n (either 0 or 1). Such a small difference between coders is mainly due to the granularity and structure of our questions and codes, unlike an open-ended interview.

5.1.4 DMCA takedown notices

To complement our survey findings we analysed how accessible the DMCA takedown process is in the 13 Android and 10 iOS most popular modded markets. The Android markets were taken from ModZoo [176] (Chapter 3), and iOS markets from iOSModZoo (Chapter 4), both lists were computed using the Tranco list [166]. We also checked Lumen [44] (a database of cease and desist notices related to online content) for notices sent by and concerning these 23 markets. Lumen mainly contains DMCA notices submitted voluntarily by Google, Twitter, Meta, etc. [44].

We focused on DMCA because it is mentioned in most modded markets and is standard practice globally. DMCA was introduced in 1998 to protect both copyright holders and online platforms hosting infringing content. It allows users to request takedown of their copyrighted materials from markets while protecting platforms from liability provided they act swiftly to remove infringing content [207]. While it is a U.S. law it has become the defacto standard worldwide because it applies to any platform that is based in or serves U.S. users. Thus, platforms operating globally or hosted (even if partially) in the U.S. must comply with the DMCA [207]. Furthermore, it has been adopted by internet service providers and online platforms globally [61, 99], and many international laws have adapted it. The European Union based their e-Commerce Directive of 2000 on the DMCA's notice-and-takedown approach [91, 200]. In 2022 the EU introduced the Digital Services Act (DSA) to update their previous directive [91, 169, 201]. Another example is the United Kingdom's Copyright, Designs and Patents Act of 1988 (CDPA) [112], which has followed the E.U. copyright laws until Brexit [172]. Some consider it inadequate for modern digital markets [165].

5.1.5 Ethical considerations

Our departmental ethics committee approved both the modded market and original app developer surveys. The surveys and research do not intend to expose participants or disclose illegal behaviour.

As the data has been de-identified, we are unable to disclose potentially identifiable information about any participants of our surveys.

Anonymised research data and records will be retained on encrypted drives for as long as needed for this study. We obtained informed consent for the surveys and advised potential participants that they should be over 18. All market operators and developers were contacted via their publicly-available contact details. Each participant received a unique URL, allowing follow up invitations, and only one response per link. The participant-URL mapping was kept separate from the survey responses and destroyed after analysis.

5.1.6 Limitations and threats to validity

The findings of our market operators survey are limited by the small number of responses, although this was expected due to the nature of the markets' activities. Previous studies have identified a steady decrease in survey response rates since the 1990s; some authors suggest mail surveys have typical response rates of 6% to 16% [72]. Studies on illicit activities have found similar and even lower response rates: a survey of darknet market sellers obtained 49 responses out of 745 (6.6%) [43], a previous study had obtained 49 responses out of 2961 (1.7%) [107] and a study on drug vendors (Silk Road) had 10 participants [211]. Studies on dark web participants had sample sizes of 4 to 17 respondents [40, 84, 143, 146, 149]. Results on these difficult to reach populations are usually published acknowledging their limitations and/or adding context using qualitative methods and measurements. We have done this in this chapter by referencing our previous results from Chapters 3 and 4 when relevant, and by studying DMCA processes in the most popular modded app markets (§5.4). Another example with a similar response rate to our surveys is a survey of developers who incurred technical debt (i.e. took shortcuts in software development to save time) in their projects, that obtained a 7.4% response rate due to the sensitiveness of the subject [145]. A recent survey of modded WhatsApp users in Kenya had 20 participants, although the authors did not disclose the response rate, only that multiple WhatsApp groups and users were contacted [153, 159].

Furthermore, with any survey there is self-selection bias, and our respondents operate relatively small and young markets with small teams. Thus, the conclusions should be interpreted carefully and might be more representative of a subset of operators.[1] This contrasts with our developers' survey that has better representation.

[1] *This paragraph is mostly personal opinion.* Regarding the modded market operator surveys, some of the iOS and most of the Android respondents operate small markets and their participation in our surveys might be an attempt at publicising their market for some of them (many asked us to include their markets' name even though we were clear from the beginning that our study would report their responses anonymously). Thus, the number of apps in their catalogue and number of downloads they report should be taken with caution. However, it is particularly worrying when they report their security testing protocols and mainly lack thereof (see §5.2.5), if we assume they were trying to paint a favourable picture of their markets. Meanwhile, bigger markets like the ones studied in previous chapters are less likely to need publicity and monitor their emails manually. The operator responses show that while many Android and iOS operators are aware that their markets are not fully legal or ethical, iOS operators tend to express more pride in going against Apple's walled-garden and helping users sideload apps. This might also explain the participation of bigger iOS markets' operators compared to Android.

Our study inherits limitations from the ModZoo and AndroZoo datasets [2, 176], including the absence of paid apps and the possibility of missing versions in AndroZoo. Our analysis is biased by the ranking of the most popular markets in ModZoo. The authors of AndroZoo have taken measures to minimise these limitations, and so did we in our ModZoo study.

The developers survey has a good representation of developers of all sizes and popularity ranks. However, many developer emails, including Google's, are no longer monitored, with developers requiring users to contact support through their apps or website forms. Others were misspelled or no longer active. Thus, our reach is limited to those with active emails who monitor them. There are also no iOS-only developers as discussed in Section 5.1.2.

We cannot compare the developer and operator responses with the ModZoo analysis results, as the operators who responded do not operate the popular markets included in ModZoo.[2] However, all the developers contacted have had their apps shared in modded markets as found in ModZoo. The effects and awareness of developers is probably mostly due to the most popular markets as analysed in ModZoo so our survey results can be compared with our previous results (Chapter 3).

5.2 Modded market operators survey

This section outlines the main findings from the Android and iOS modded app market operators surveys. All quotes are given verbatim. The Android modded market operators survey was completed by 14 modded market operators (4% response rate). Most operators reported running more than one market, thus the responses cover the operators of at least 19 markets. Five further responses were received via email, including operators providing a summary of the answers to some of the more relevant questions from the survey. Some responded to say they did not see a benefit in answering the survey, others only provided automated responses listing article and ad publishing prices. The responses are biased towards smaller operators and there is a significant self-selection bias inherent in surveys like these, as discussed in Section 5.1.6. The relatively small response rates are also discussed in §5.1.6.

The iOS modded market operators survey was completed by five (6% response rate). Two other market operators who did not want to complete our survey answered our email. One said their market is an illicit market for iOS apps and was aware that Apple does not allow app redistribution. The other operator stated that their market is a distributed app store in which they host apps themselves while also allowing other developers to host apps.

Respondents of the Android survey are numbered A1–A14 and the iOS respondents were numbered I1–I5, ordered by the number of apps in their respective markets so that A1 contained the most

[2] *This paragraph is mostly personal opinion.* We would like to think that if the most popular markets Android modded app markets had responded, the results on security measures would have been different. In fact, two very popular sideloading Android markets also responded to our survey. They do not focus on modded apps and are thus not included in this study, but their answers showed they have security testing mechanisms in place, including periodic scanning of apps. However, Chapters 3 and 4 showed the most popular modded app markets are less secure and host a lot more malware compared to the official markets.

apps, and so on. We note some iOS markets host a few apps developed by themselves and mainly third-party repositories, and thus their focus is different in terms of vetting those repositories and the apps they host. Many also offer tools to sign third-party apps via user credentials, taking advantage of Apple developer tools.

5.2.1 Android modded markets operation

All Android market operators started their markets after 2015, many around 2020. All are men, mostly aged 25–34, while a few are 18–24 and 35–44 years old. They operate from Asia and Africa, and most consider their markets their main occupation, except for two bloggers and two students. They reported spending 7–120 h/month maintaining their markets. Some run them alone but most have 1–3 employees. None are involved with other online services, except A14, who provides website development services and blogging.

Operators A1, A10 and A14 wrote their own code to set up their markets, while A5, A7, A8, A11, A12 and A13 used existing tools or code they modified to develop their market platform. Operators A1, A7, A11 and A12 do not advertise their market. Operator A5 advertises on social media and search engines, while A5, A8, A10 and A14 link to their markets on social media platforms: A8 on Telegram, A10 on Facebook, Pinterest and Instagram, operator A13 reported using SEO and advertising in related forums, and A14 used "Third Party App Like Whatsapp" and also tried running Facebook Ads but were always declined. Nearly all operators reported most of the visitors that land on their platform arrive through search engines, while A1 and A7 reported the majority of their visits come from direct visits. The majority stated this was confirmed by Google Analytics. A13's market receives 1–2 million visits per month.

Operators A1, A5, A6, A8, A10, A12 and A13 collect apps and release them on the platform themselves, using tools to download the apps directly from Google Play. Furthermore, A12 compiles open-source apps from GitHub. A7, A11 and A14 get apps from other (unreported) websites. Operators A1, A5 and A13 add a large number of apps (> 800) to their market on a monthly basis, while A6, A7, A8, A10, A11 and A14 reported much lower numbers (< 20).

As well as collecting apps from Google Play, A5 is contacted by developers wanting to promote apps through their market, and requests developers to submit basic information like version, size, minimum requirements and description before submitting it to the market. None of the market operators reported having been contacted by an original app developer about bad reviews on their site. Only A6 and A7 admitted re-signing and modifying apps and their binaries and/or metadata before releasing them in their markets, another 9 operators responded they do not, and the remaining three did not respond to this question.

5.2.2 iOS modded markets operation

Our iOS market operators survey revealed most respondents started their markets after 2019 and are operated by men aged 18–34. However, I1 started their market around 2012 and I2 around 2016.

The market operators reside in Europe and North America, and operators I2, I4, and I5 consider the operation of their iOS market as their main occupation. They spend an average of 97 hours per month (15–180 h/month) maintaining them. Two of the operators work alone; the other markets are run by teams of two, five and more than ten people. One operator declared their market's "APIs are free and open for everyone" and that lots of apps and services use them. Another ran a jailbreak repository before starting the market.

As mentioned before, some stores provide third-party repositories and tools to sign third-party apps. I5 is one of them, stating: "This stores goal is not to provide a large library of apps it's to give the user the ability to install their own apps". Most operators wrote their market websites and apps from scratch. Operators I1, I3 and I4 do not advertise their market but iOS modded markets typically have social media accounts. I5 added that "sometimes people on TikTok are hired to make videos on the product which is how other stores usually advertise their services too [...] - through indirect advertisements are how we grow as large as we do".

Operator I1 claimed over 220k active user devices in 6 months, and they estimate (as they do not use analytics) their website gets two-thirds of their visitors from links on other sites and direct visits. I2 does use analytics, with the majority of visitors arriving at their site from search engines, less than a third from links and a tenth visit the site directly. Operator I3 estimates three-quarters of visitors come from their associated Telegram, Discord or similar groups or channels. Operator I4 estimates roughly 90% of the visits to their website come from links on other sites and the rest from search engines, as they "definitely don't do any marketing". All operators declared that none of their visitors come from advertisements except I5, who estimates that their market receives 70% of visitors this way. Operators are answering mainly about their websites, but I1, I4 and I5 also have an iOS market app, to which all visits are direct visits.

Operators I1 and I3 allow users and/or developers to submit apps to their platform, including anonymous app submissions. I1 states around 2 000 developers while I3 reports only three developers or users submit apps every month to their market. Meanwhile, operator I4 develops their own apps while also allowing distribution by third party developers. Operators I2 and I5 collect apps themselves by building them from source code available in GitHub.

In terms of apps added to and updated in their markets in an average month, I1 reported 30 000 apps added and 300 updated, I2 reported 55 and 50, respectively, the rest fewer than 10 in both categories. In terms of downloads per month I5 reported the most (10 000 000), followed by I4 with 300 000, I1 with 170 000, I3 with 45 000, and finally I2 with 1 000 app downloads. Operator I4 added their market is distributed by design, and every developer hosts their own apps. As they do not have analytics for all apps distributed through their platform, their answers of added and updated apps reflect their first-party apps, while the download count does cover all apps across all sources.

Third-party developer uploads were explained in detail by operator I3. Developers can upload IPA files in the platform, and an in-place automated system parses the information property list (Info.plist) file for relevant metadata (name, file size, icon, dynamic library files (.dylib), etc.) present in the app. Operator I4 asks developers to upload the app to a public URL (personal website, GitHub releases,

etc.) and then create a metadata file with the relevant information (name, description, versions, the download URL, etc.) conforming to the specifications provided by the market. Once the developer uploads the metadata, the app is shown in the market alongside the market's first party apps and apps from other sources the developer has added. Unlike Android markets, most iOS markets we identified provide the app metadata and a link to public hosting platforms to obtain the apps. Only I1 hosts user comments in their market, but no markets have any rating systems. All market operators state they do not modify the apps before releasing them on their platforms; I4 states their installation process requires changing app IDs to something unique since their sideloading method (personal, free Apple developer accounts) does not allow multiple users to install apps with the same ID.

5.2.3 Android operator motivations, monetisation and intellectual property

We ask operators how they became involved in running these markets, their current motivations, and their short and long-term goals. Technology has always been operator A1's passion. He wants to increase interest in Android apps and games through the market, with the goal of always keeping it up-to-date. A5, who is also an Android developer, considers this their profession and aims to earn money while offering new apps and games, especially those not available in Google Play. For A7 the main motivation are the users and offering them apps that are not available in Google Play, their goal is to become the largest third-party app market. Operator A10 believes developers can get more installs and traffic when listed in their market in addition to Google Play. A11 wants more visits in the blog posts part of their market, while A12 aims to become one of the best and biggest third-party app markets. Operator A13 wants to grow their platform, make it more secure and give more control to developers. In terms of future goals, adding more apps to keep their platform updated, and making more money were the main factors, with only A13 mentioning making their market more secure.

In relation to monetisation, operators A1, A5, A10, A11, A12 and A13 report using ads for monetising the market with four of them specifying 'Google AdSense' as one of their providers. They and A14 reinvest less than half of the market's income in platform maintenance. All respondents reported half or less of their total monthly income is generated from their market.

In terms of intellectual property, operators A1, A5, A7 and A13 stated having been contacted via email and their DMCA forms by developers claiming their apps had appeared in their markets without authorisation. A1 and A5 reported these were finance and banking apps which they permanently removed from their markets, although A5 specified this only happened with one developer, and they removed the app within 48 hours. Similarly, A7 and A13 also removed the apps once they were contacted. When asked about legal compliance, only operators A1, A10, and A13 reported having concerns when operating their markets. Meanwhile, operators A5 and A11 reported having no concerns.

5.2.4 iOS operator motivations, monetisation and intellectual property

Operator I1 states they are the largest and oldest (third-party) IPA database, created after a previous platform was taken down by its creator due to growing piracy in it. I3 ran a successful jailbreak repository (a way to distribute apps and 'tweaks' to jailbroken iPhones), and took the 'logical' jump to serving non-jailbroken users IPAs. Operator I4 started after developing an emulator which was rejected from Apple's App Store, motivating them to build their own market to distribute their work and help other developers in similar situations. I5 started after learning the trade from the creator of an early iOS third-party store. All operators state modded apps are not their market's main focus, with I2 denying the presence of modded apps in their market. Operator I4 does not provide the modified apps themselves as they are a distributed app store and developers offer them directly in their platform.

In terms of their future goals, I1, I2, and others want to keep their community and accessibility, I1, I3, and I5 also want to improve the user experience and stability. Furthermore, I3 wants to monetise premium uploads and their app signing service. Meanwhile, I4 wants to build a "complementary store to the App Store that offers unique, quirky apps that aren't allowed in the App Store for various reasons (emulators, virtual machines, clipboard managers, game streaming apps, non-Swift compilers, etc.). [...] we do NOT want to compete with the App Store — we want to co-exist".

In relation to monetisation, I2 and I5 rely on advertisements to monetise their markets, while I4 relies on donations through Patreon. I1 and I3 do not monetise their markets. I1 considers the market to be a non-profit project, stating premium subscriptions pay for servers and other expenses only. I4 and I5 report more than 80% of their monthly income is generated by the market, while I2 reports a low percentage. I3 and I5 report less than 10% of the market income is reinvested in platform maintenance, while I2 reinvests around 35%, and I1 and I4 a very high percentage.

Regarding developer consent and intellectual property, all market operators except I1 deny having been contacted by developers claiming their apps were being hosted without consent. Operator I1 was contacted under the DMCA, but no action was taken because "[our market] is registered in kingdom of Tonga, where there is no such legislation". This is a misconception since this content is accessible to users worldwide including in the EU and the U.S., which makes them liable under DMCA as discussed in §5.1.4. Only I1 and I4 report having concerns about legal compliance when operating their third-party markets. However, I1 considers their market to be fully legal due to not hosting any apps, being "only a metadata engine and IPA signing service" because the apps (IPA files) are hosted in third-party file hosting sites. I4 added that their responses would be much different once the EU's Digital Markets Act was enacted and Apple had to officially allow sideloading. Only one modded market besides AltStore [198] has tried to become an official third-party market through Apple's interoperability program [25] after the DMA got enacted without much success [11, 13, 17]. Although not concerned, I5 were given legal advice before starting their market.

RQs1: What motivates operators to run modded app markets?

Operators have an economic incentive to break copyright law, some are motivated by making sideloading accessible to users and becoming a big market. Small Android market operators in this sample reside in Asia and Africa, consider this their main occupation even though it generates half or less of their total monthly income, and mostly monetise their markets through ads. The iOS operators in this small sample reside in Europe and North America, and three of them consider this their main occupation. A few use ads to monetise their markets, others do not monetise it and another relies on donations. One iOS market had 220k users in 6 months, the biggest reported 30k apps added per month, and the smallest reported 10 million downloads/month.

RQs2: What are modded market operators' revenue streams?

Android operators of markets of all sizes report using ads to monetise their markets, but all of them reported less than half of their monthly income comes from their markets. All iOS market operators stated modded apps are not their market's main focus and some of them run distributed app stores with developers adding directly to them. The biggest iOS market is a non-profit project, the third biggest also does not monetise it, and the second and two smallest rely on ads. The latter generates most of the operators' monthly income, while the others generate a very small proportion. Only four Android markets report contact with developers through DMCA notices, and claimed they took immediate action. One iOS operator was contacted but took no action as they provide links to third-party file hosting sites and their market is registered in Tonga.

5.2.5 Android operators security measures

We ask market operators about the steps and security measures they follow before publishing apps in their markets. They perform very little testing beyond one platform using VirusTotal. Operators A1 and A7 scan apps using security tools. Other operators simply ask developers for information about the app, do "background checks", do manual reviews or testing before uploading them to their markets.

For apps already in their markets, A1, A5, and A7 perform manual testing and use antivirus scanners. There are many misconceptions regarding Android and Google Play app security. A1 states "playstore apps are already secured". Similarly, A10 also believes in the security of Google Play so does not perform any security testing. Operator A11 also performs no testing: "we just get the app from trusted websites and upload it to our site". A13 only uses "credible sources for apks", and A14 states "I Use Android Device, So I Prefer The Google Play Protect" as a reason for not performing any security testing. No operator mentioned any long-term security procedures beyond the initial upload, except A12, who has "a team that does regular check and background check on apps download from our site", without specifying whether this includes any security testing or just compatibility, usability, etc. All operators remove apps after users report security flaws, except A11 who had not received any reports.

5.2.6 iOS operators security measures

In terms of security protocols and measures, I1 states apps are tested by the uploaders (unspecified methods), combined with a reporting system where "people can report unwanted activities of an app". They reported no issues and not having changed their security testing methods over time. However, I1 has recently changed methodology to now automatically running antivirus checks on every app uploaded to their market. Operator I2 states they "get the apps from the original source" and they scan them using VirusTotal. Operator I3 has no security testing, with "a small group of trusted people with uploading privileges". Operator I4 manually reviews apps in their "Trusted Sources", with no additional testing yet for apps from third-party sources or downloaded from the internet (which users can sideload using their method). They are currently beta-testing the "untrusted" sources feature, with access limited to patrons. Finally, I5 believes "All apps on iOS are sandboxed meaning they can only access the contents of its own container and bypassing this means the device is already exploited to begin with". They add that "Any important information is stored by Apple on a separate processor called SEP since around 2010", this is not quite true.[3]

We ask what steps are taken when a security flaw is reported. Operators I1, I2 and I3 remove the app and the security of the app is verified internally. Operator I1 also removes the app from user devices. As I4 mainly hosts their own apps, they "Prioritize the feedback and immediately try to develop a solution, and release a fix [...]". For third-party apps, they also "have the ability to remotely block Sources in case a Source turns out to distribute malicious/dangerous apps", although this is "primarily intended for extreme cases". Again, I5 believes no security flaws are ever present in apps due to Apple's iOS security mechanisms, plus the mobile provision file which only Apple can create and limits the *entitlements* an app can use.[4] Operator I5 added their servers have "been attacked by many competitors" requiring them to strengthen their security measures.

RQs3: What types of security testing do they perform, if any?

Android operators perform little testing: only the biggest and two middle Android markets use available tools to perform simple app scans. Similarly, all iOS operators perform limited or no security testing on the apps, however they remove potentially problematic apps after user reports. There are many misconceptions regarding Android and Google Play app security, and sandboxing and iOS security.

[3]The Secure Enclave Processor stores device passwords, fingerprints and facial features needed for face recognition, and is "designed to keep sensitive user data secure even when the Application Processor kernel becomes compromised" while also generating keys for applications when requested [35]. However, apps are not allowed to store data in it, nor is it mandatory for apps to use keys generated by it to store their data. Thus, the personal data, login details, access to cameras, microphones, etc. that an app may have are not protected by the Secure Enclave and malicious modders or developers could still extract user data through modded apps. There are other mechanisms provided by Apple for security purposes including app sandboxing, the Data Protection API [22] the Keychain Services API [24], and others that used properly would limit the damage that any one modded app could do and the data they could access. However, to reiterate, they cannot prevent malicious actors from extracting data from the app they have modded if that was their intention.

[4]Apps hosted in these markets are not signed by Apple so this does not hold.

Table 5.2 Distribution of developers by (Android) installs

Installs	Developers contacted	Respondents
≤100	375	24
101 – 1 000	869	35
1,001 – 10 000	2 094	83
10 001 – 100 000	3 682	136
100 001 – 500 000	4 082	108
500 001 – 2 000 000	4 865	129
2 000 001 – 10 000 000	6 116	124
≥ 10 000 001	5 704	78

5.3 Original app developers survey

This section outlines the main findings of the first survey-based study of the effects of modded app markets on original Android (and iOS) app developers. All quotes are given verbatim.

A total of 717 original app developers answered our survey (2.60% response rate). We received further email responses from developers asking for more details and clarification to confirm it was not a phishing attempt or spam. Others were customer support agents who forwarded it to the correct teams, while many were simply automated emails with a customer help ticket number or a form to fill. We also received emails indicating the email address is no longer monitored or a no-reply address, meaning support is handled within the Android (and iOS) app directly.

We divided the respondents into four quarters according to the total installs of all their apps combined. Thus, Q1 covers developers 1 to 179, Q2 180 to 358, Q3 359 to 537, and Q4 538 to 717, where developer one has the most installs and developer 717 has the least. This gives us some insights of how different groups of developers are affected by different aspects of modding. No quarters are mentioned in the following sections where the distribution is even.

5.3.1 Respondent 'demographics'

The 717 responses are from a representative sample of the original app developers both in terms of total app installs and popularity rank within those invited to participate, as well as team size and other aspects, as discussed later. The respondents have an average rank of ∼16 800 out of ∼27 700 based on their apps' number of installs, with a median of 17 844 and a standard deviation of 7 587. These developers have a total of 4.87 billion installs with an average of ∼6 825 000 and a median of ∼330 500 installs. There is a good distribution of respondents (developers) of all number of installs and rank, as shown in Table 5.2. Furthermore, the distribution is similar between developers contacted and respondents in each bracket. A more detailed view of the respondents' number of installs and ranks are included in Figures 5.1 and 5.2, respectively.

In terms of company size, respondents reported four developers on average and a median of one developer. The largest was 60 employees. As most were solo developers, four in five respondents

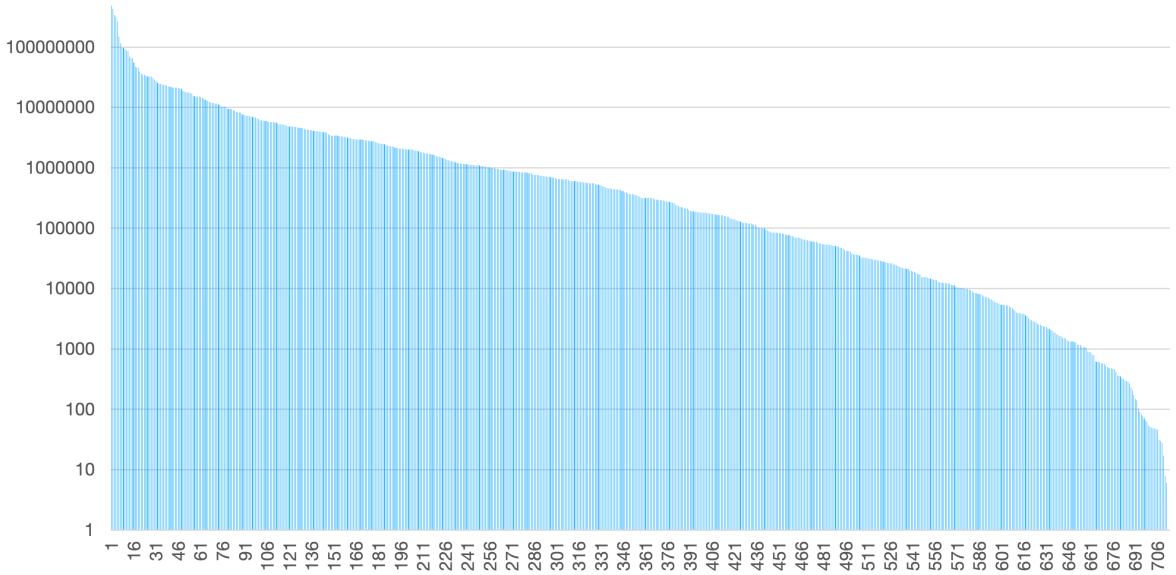


Fig. 5.1 Android installs of the 717 respondents in logarithmic scale

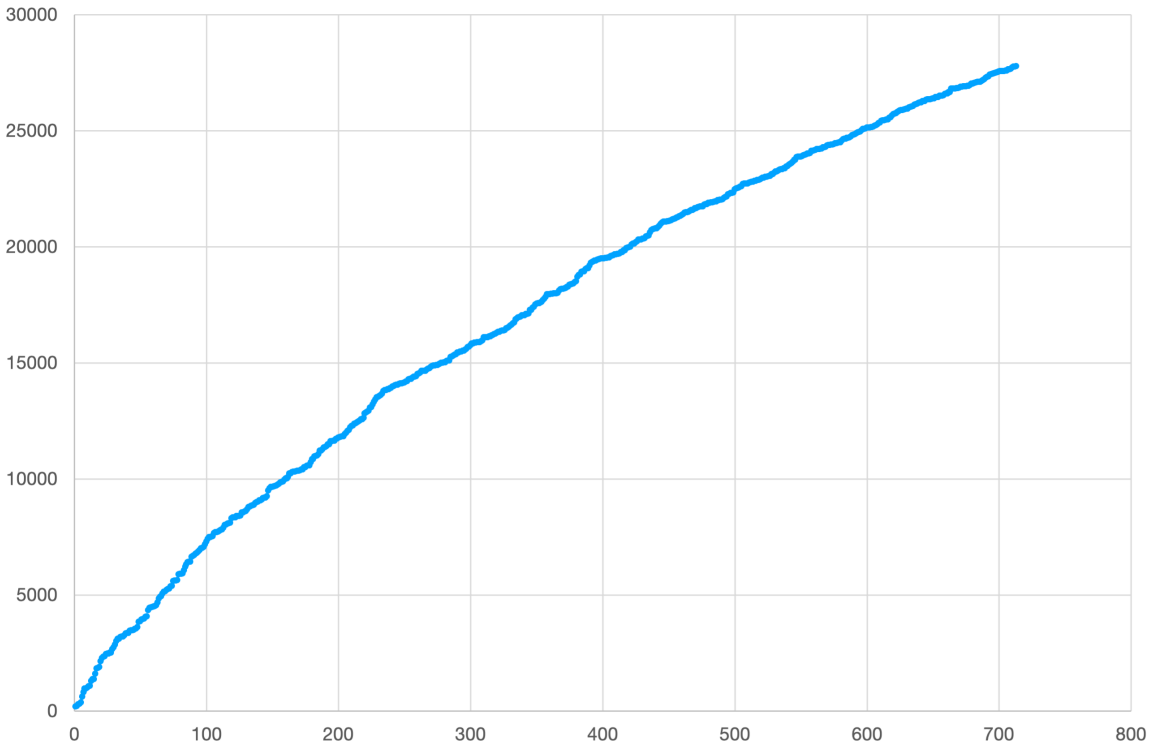


Fig. 5.2 Respondent overall rank based on Android installs

reported being "CEO or similar" or solo developers and founders. Others were mainly "lead developer" and "tech leads", with some senior developers and a few junior, and medium-level developers.

Almost all respondents develop apps for the Android operating system, except for a handful whose apps are no longer in development. This is unsurprising, since we obtained their contact details from their publicly listed emails in their apps' Google Play pages. However, 54.3% also develop iOS applications, and 6.3% also develop for other platforms (mainly Windows PCs, Web, gaming consoles, macOS, and also Huawei, visionOS, Linux, and smart TV apps).

RQs4: Which developers have had their apps modded?

Developers of all sizes have had modified versions of their apps and games of all categories appear in modded markets, as studied in previous chapters (Chapters 3 and 4). We obtained 717 responses with an even representation across the spectrum of installs and rank of the 27k contacted. Respondents have a combined 4.87 billion installs in Google Play with an average of 6.8 million, most were solo developers, and more than half are also iOS app developers.

5.3.2 Developer awareness of modded apps and markets

Most developers knew of Android modded markets, fewer of iOS markets and sideloading. Of those who answered this question (500+), 81.0% were aware of the existence of modded Android apps, and 74.9% were aware of modded Android app markets. However, only 39.4% and 34.5% were aware of modded iOS apps and markets, respectively. A majority (63.0%) had noticed their own apps in a modded app market before our emails. Developers had noticed [5.28, 2] (average, median) of their own Android apps in these markets, versus [2.28, 0] iOS apps in modded markets.

Almost all (91.8%) developers confirmed they had not authorised these apps to appear in the markets they were found in ModZoo [176] (out of the 391 who answered this question). The remaining 8.2% that authorised their apps to be included in the markets are mainly open-source projects and apps. Thus, they did not explicitly authorise their apps to be modded, but implicitly authorised this use as part of open-source software terms and conditions.

RQs5: Are developers aware of modded app markets?

More than 80% of developers in our survey were aware of modded Android apps and markets, but less than 40% were aware of modded iOS apps and markets. Most had seen their own apps in modded markets before. Only 32 developers had authorised their apps in modded markets.

5.3.3 Developer revenue losses

Most developers reported incurring revenue losses from in-app purchases (54.5%), advertising revenue (45.2%), app sales (38.7%), and others (12.7%). Those that included estimations of losses (in USD) for each of the first three categories declared an [average, median] of: [1 669, 1 000] in in-app purchases, [1 451, 590] in advertising revenue, and [2 077, 1 000] in app sales.

Elaborations (free-form text) on aspects of revenue loss were provided by 292 developers: 68.8% incurred significant losses of different types (discussed below), while 17.5% incurred a small revenue loss (mainly the Q4 developers) and 13.7% could not make an estimation. This section, including the numbers above, should be interpreted with caution. Most developers (80.1%) made estimations based on rough guesses and the little information available to them. Only 19.9% provided estimations based on data (typically analytics and other tools discussed in §5.3.6).

5.3.4 Modifications found by developers

A total of 251 respondents elaborated further (open text) on modifications found in the modded versions of their apps. Ten developers did not find any modifications in the modded app market versions of their apps, confirming observations from the ModZoo project [176] and the market operators' survey that some apps are copied from Google Play and hosted in the markets unchanged. Of the developers who discussed modifications, 132 did not perform any analysis of the modded apps. Reasons included lack of time and resources, not trusting the source, and thinking they pose security risks. Others think there is nothing practical that can be done against these modders. A further 2% did not care about changes due to the open source nature of their apps.

Of the 114 that analysed the modifications to their apps, 87 (76.3%) found feature mods: 60 (69.0%) found premium versions or features included for free (90% are from Q1-Q3), and 22 (25.3%) found cheats (82% are from Q2 and Q3). A further 32 developers (28.1% of the 114 who checked) found changed advertising in their apps: 65.6% found advertising had been removed (48% of them from Q1 and 29% from Q2), while 31.3% found added advertising, and two (6.3%) found their advertising IDs had been t—a practise reported in ModZoo to affect more than one in five apps that include ad IDs [176]. Fourteen developers (10 of them Q3 and Q4 developers) found protections had been removed from their apps: half had removed license checks, others had disabled server checks. Five Q4 developers (4.4%) found their paid apps had been pirated and uploaded to the modded app markets. Finally, two developers found modded versions of their apps had added permissions, and another two found their contact information had been replaced.

One developer added: "Most of the apps analyzed had basic code changes: adding blocks of code to bypass the store's license verification system. However, there were more extreme cases where permissions were added. Including permission to overlay apps, showing ads when the app was running in the background. In the most extreme case detected, there were accessibility permissions (screen reader), and obfuscated native code (C/C++), which could possibly contain some form of malware." This encapsulates the main types of modifications modders can make to their benefit that go beyond what modded app users would expect to be modded such as premium features or less ads.

5.3.5 Other impacts of modding

Modding has many impacts on developers beyond revenue loss, with 345 respondents (48.1%) elaborating further on these impacts. No further impacts other than revenue losses were reported

by 224 (64.9%), due to having a small user base or offline only apps and games. Another seven (6.1%) (four of them from Q4 and none from Q1) perceived positive effects of modding, mainly more exposure and online discussions.

In terms of the various negative impacts that 114 respondents experienced, the most common ones are (with percentages out of the 114 who experienced negative impacts): 39 (34.2%) experienced increased server calls and the associated increased cost in most cases; 32 (28.1%) received support requests for modded versions of their apps or features errors; and 16 (14.0%) noticed more errors and crashes. Developers claimed these can not only damage their reputation, but also waste their time. There were complaints about the time wasted in trying to replicate and fix errors before realising it was a modded app error and not their base app.

The impact on other users was mentioned by 18 developers (15.8%), including 11 (9.6%) that specified this was due to unbalanced online matches. This mainly affected Q3 (55%) and Q2 (27%) developers. Fourteen developers (12.3% and eleven of them from Q1 and Q2) mentioned suffering reputational damage for all of the reasons mentioned above including impact on genuine users, app crashes, malicious ads and malware. Another seven (6.1% and five of them from Q2 and Q3) mentioned the increased cost of third-party services inside their apps resulting from modded versions making use of them. Four developers worried about the security of their apps' backend and machine learning models, three developers felt demoralised and one of them even demotivated to implement new features as a result of modded versions of their apps. All of them were aware before we contacted them of: modded android and iOS apps, modded Android and iOS app markets, and even of the presence of their own apps in modded apps markets. Thus, this demoralisation was not a result of our contacting them. We hope to have helped developers by bringing this situation to their attention. Furthermore, 13 developers thanked us for contacting them, giving them a list of markets, making them aware of this issue, or making others aware of it. All of these were unprompted responses included in the free-form text boxes in different parts of our survey.

RQs6: How do modded and pirated versions affect developers?

Developers of all sizes reported revenue losses and found their apps were modified to include new features and subscriptions or with changes made to advertising identifiers. Typically the bottom quartile (Q4) developers reported smaller losses. Premium features were added mostly to Q1–Q3 apps, and removed advertising particularly affected Q1 and to a lesser extent Q2 developers, showing the demand for popular apps without ads. Developers faced increased costs and support requests. While reputational damage affected bigger (Q1 and Q2) developers the most, middle (Q2 and Q3) developers noticed the increased costs from the non-paying users the most. Some smaller developers gained exposure while others were demoralised.

5.3.6 Prevention and detection techniques

We asked developers to answer in free form text about any protection techniques they use in their apps. This included prevention techniques to combat modding, and also detection techniques to quantify

the issue or block illegitimate requests, users, etc. Prevention was discussed by 439 respondents, of which 232 or 52.8% do not use any prevention techniques (33% of them Q4 developers). For the other 207, the most commonly used techniques are: Obfuscation (80 developers, 38.6%); ProGuard [106] (32, 21 from Q1 and Q2); and DexProtector [137] (2, both from Q1). Meanwhile, 66 developers (31.9%) use a proprietary custom made method, including a mix of other techniques and custom checks. Server-side checks are used by 42 developers (20.3% of the 207, 64% of them from Q2 and Q3) to thwart the functionality of a mod by checking legitimate purchases have been made, player stats are correct, etc. However, they are also known to limit the user experience and require an internet connection.

In terms of prevention techniques offered by Google, 24 developers (11.6%, 17 of them from Q3 and Q4, only 2 from Q1) use Google Play Licensing [6] and 19 (9.2%, only one from Q4) use Google's Play Integrity API [8] to prevent unofficial copies of their apps from running. Prevention techniques can cause problems to legitimate users as well, according to 15 developers (7.2%). Finally, seven developers use third-party prevention tools (most from Q3 and Q4), while four said third-party tools are expensive. Fifteen developers (2.1% of all respondents, only one from Q1) consider they lack the technical knowledge required to combat modded apps.

Detection techniques were discussed by 396 respondents, with 303 (76.5%) not using any detection techniques (67% of them Q3 and Q4 developers). For those who do, the most common were: Proprietary detection methods (13, 11, 10, and 7 developers, from Q1 to Q4 respectively); server-side checks to detect modded versions (25 developers); and a third-party tool (7), including two that use DexProtector and other Runtime Application Self Protection (RASP) tools. And finally four developers check the installer package [7].

5.3.7 DMCA-related actions

DMCA (Digital Millennium Copyright Act) takedown notices are the most prominent way to request and enforce the removal of copyrighted content, as discussed in §5.1.4. When asked, 54.7% of developers said they will not request takedowns, versus 45.3% that would. Of the 426 developers (59.4% of the total survey respondents) who elaborated further, 420 (98.6%) discussed previous actions they had taken, while only 6.8% (29) mentioned future actions they will take. In terms of previous actions, only seven developers (1.7% out of the 420, none from Q4) report having had success and 37 (8.8%) limited success with DMCA notices. Most (349, 83.0%) have never submitted DMCA notices before, with 3.7% (13, 54% of them from Q4) unaware of modded markets before our survey, and 2.9% (10) unaware of DMCA requests. Only 72 developers (17.1%) had tried requesting the removal of their apps, of which 65.3% (47) report no success with DMCA notices in several occasions. This is mainly due to: receiving no response (66.0%, 31), the process being very time consuming (12.8%, 6), and/or DMCA forms requiring too much proof of ownership or personal information (8.5%, 4 developers). Only 51.4% (37 developers) report limited success with DMCA notices in some occasions, with 18 of them (almost half of them) having seen their apps reappear

after their request (typically after the next version of their app was made available in Google Play or modded). Furthermore, two developers resorted to cease and desist notices and five (6.9% of those that used DMCA notices) requested removals through third-party services. Another eight developers requested Google Search to remove links to modded markets sharing their apps from its results. Also, two developers said the markets threaten to take legal actions if the person making the request could not prove their authorship of the apps beyond any doubt.

Of the 29 developers who mentioned future actions, only 48.3% (14) will be trying to request removals through DMCA notices. The other 18 developers (62.1%) will be adding different types of protections and checks against mods. One developer summarised the problem with the current DMCA notices: "I believe this is a soft rather than a hard problem. The solution should be legal rather than technological. Unfortunately that's difficult enough in just one jurisdiction, not to mention the global internet. DMCA takedown requests are pretty toothless the way they work today. At most they'll get one version of a modded app removed from one unauthorized app store for a while. To actually get any more than that done (or if the app store simply refuses the request), the developer has to go to court over the issue, which is often out of reach for smaller developers with little resources. If I tried to do that I'd likely have to pay much more in legal fees than I'd ever stand to recover, and it might not even be successful".

5.3.8 Current solutions by Google and Apple

Google and Apple have measures currently in place. Since Google allows sideloading by default in Android, they also provide developers protection and security libraries and tools to prevent modding and copyright infringement within and outside their app store. When asked about these and what else both companies could do to help developers against modded apps in free form text, more than 75 developers discussed current solutions and tools. Twenty were unaware of current solutions provided by Google and Apple to defend against app modding. Protection tools provided by Google were mentioned by 56 developers. Of the 27 mentioning the Google Play Integrity API (41% of them from Q1, 30% from Q2), almost a third stated it did not work (as evidenced by us contacting them about their modded apps or as noticed during their own checks). Of the 11 developers mentioning Google Play Licensing, seven stated it does not work.

RQs7: What tools are available and used by developers?

Popular tools and techniques include obfuscation, third-party tools and server-side checks. More than half of the developers do not use tools or techniques to prevent modding, a third are Q4 developers. Similarly, most developers (77%) do not use detection techniques, most of them are Q3 and Q4 developers. Developers are more likely to find third-party tools more expensive and lack technical knowledge to combat modding the smaller the developers are. Many were unaware of tools offered by Google, and many stated they do not work against modding.

5.3.9 Possible solutions by Google and Apple

More developers discussed possible future protections and solutions by Google and Apple: 143 developers (19.9% of all respondents) believe Google (and some also Apple) should provide more tools or techniques to protect against modded apps and sideloading. Eighteen respondents (2.5% of the total) explicitly stated that Apple does a better job at tackling sideloading and modded apps, with five of them adding they had never encountered modded versions of their iOS apps. Furthermore, nine respondents believe Google should not allow sideloading and follow an approach similar to Apple's. Another 77 developers (10.7% of respondents) think Google (and Apple) should not provide any tools or techniques against modding. Of these, 26.0% clarified they hold this view because they believe any well-known technique would likely be thwarted due to the intense desire and benefits from app modding by users and modders.

A DMCA solution from Apple and Google would be useful according to nine developers, including a tool to help request takedown automatically. Thirteen developers would file DMCA requests if they were automated or made easier. Suggestions included easier automatic finding of modded apps infringing their copyright and generating a dashboard or report for the original developers. Another six respondents think search engines like Google should protect copyright holders, two wanted to see trademark protections, such as automatically hiding modded markets from search results.

5.3.10 Combating modded apps and piracy

At various points in the survey, 221 developers expressed unprompted beliefs related to combating modded apps and the future of piracy. The following beliefs were entirely negative:

- Nothing can be done about modded apps and markets, and it is not worth the hassle according to 132 developers, 18.4% of the total. They believe:
  - Contacting the markets for removal is useless (48, 36.4%; 73% of them from Q3 and Q4)
  - It is impossible to keep up with modded markets (40, 30.3%)
  - It is particularly difficult for smaller developers (34, 25.8%)
  - Modded apps are accepted as part of the Android platform (14, 10.6%)
  - Trying to submit DMCA requests is more expensive than accepting the situation (13, 9.8%).
- Piracy will always exist (77, 10.7%; 50 from Q3 and Q4)
- Modded apps entail risks to users (20, 2.8%)
- Modded app users deserve the risks associated with modded apps (4).

A minority expressed neutral or less pessimistic views:

- Modding is not a big deal (69, 9.6%; 68% from Q3 and Q4)
- Modded app users would not pay anyway (41, 5.7%)
- Modded apps are good for business and reach (21, 2.9%)
- Users will buy their apps if they like them enough (12).

RQs8: Are DMCA notices effective and used by developers?

Smaller developers were more prominent among those unaware of modded markets and those unsuccessful with DMCA requests. They were more likely to feel powerless against modded markets and piracy, but many of them also consider modding good for business and reach for smaller developers. DMCA requests are not functional at the scale of these modded markets, and are inaccessible to smaller developers as any meaningful result requires sending constant written requests for each version of each app in a market and/or legal procedures.

5.4 DMCA takedown notices in modded markets and search engines

As outlined in Section 5.1.4, we analysed 23 modded markets in detail to identify how they presented DMCA-related processes (§5.4.1). We also looked at the DMCA notices sent to search engines that involved each of the modded markets in Section 5.4.2.

5.4.1 DMCA in modded app markets

Some markets are unreachable, as we know from the 18 iOS markets that do not provide any contact information, and the 22 undelivered emails and 4 automated email responses we received from the Android market operator survey invitations. Two Android markets provide no contact information, one claims to not have changed any in-app advertising but states by downloading apps users "agree that the applications may contain malicious, advertising or virus code that: 1. may harm your device to some extent. 2. incur unforeseen, not previously agreed, financial expenses on your part".

Meanwhile, others have contact information but try to deter developers from using this function. Two markets provide DMCA pages with a form or email and state the required information but also use menacing language stating "you will be liable for damaged (including costs and attorneys' fees) if you misrepresent information listed on our site that is infringing on your copyrights. We suggest that you first contact an attorney for legal assistance". A third has a similar text on damages and seeking legal advice, and warns against contacting ISPs, as it could slow down the process. Another also mentions the liability of false claims, and adds they will ask developers for confirmation through email before deleting the content, and that "In the absence of a response within one working day, we reserve the right to continue to host the application", placing an unnecessary burden on copyright holders. Another has a DMCA page titled "Denial of responsibility" and their "Copyright Holders" page asks developers to "please use the link Feedback, which is located in the basement of the site".

We located the relevant email in a submenu. They require developers to "Send scanned documents [...] confirming the copyright of the application" and "On the official website of the app [...] you must create a .txt file named *[market]*, and then send a link to it to the site administration".

Six markets provide their email for contact in a separate DMCA Report site and one in the site footer. However, one tries again to dissuade users from contacting ISPs: "emailing your complaint to other parties such as our Internet Service Provider [...] may result in a delayed response due the complaint not properly being filed".

Similarly, five iOS markets provide a contact email to submit DMCA notices. One is only shown after clicking a link that opens an unrelated tab filled with ads or automatic downloads similar to the rest of the site. They also "do not monitor, review or analyze in any way the files uploaded to our servers by our service users. We take copyright infringement very seriously and will vigorously protect the rights of legal copyright owners". Another two markets provide DMCA forms, one requires developers or their representatives to pass a CAPTCHA before sending their personal information, passport number etc. as part of the DMCA notice. The other urges developers to "[...] create a *[market]* Publisher account to publish and monetise your app [...] and use our content tagging system that automatically prevents users from uploading of unlicensed copies of your app". Another market does not mention DMCA specifically, as they only host apps they themselves have written and third-party ones after agreeing with developers directly. However, they have privacy policies and emails developers can contact for inquiries.

Others are more cumbersome, one provides an email for other inquiries, but urges copyright owners to submit claims "here", with no form or link present in the page. They also state "The designated email *[redacted]* should ONLY be contacted for the other queries related to intellectual property right issues rather than the takedown notices. Takedown notices sent to any *[market]* email address may be delayed or ignored". Two markets have only generic contact emails with no mention of copyright or DMCA.

Finally, another does not provide any DMCA notice form or contact details besides joining their Discord server or following them on Twitter despite stating "While we believe in freedom, we also respect the rights of creators. *[Market]* strictly refrains from piracy, ensuring a fair and ethical digital environment".

5.4.2 DMCA notices to search engines

Besides contacting the modded app markets directly, with the problems that entails as explained by our survey respondents and discussed above, developers and their legal teams can also contact search engines to prevent modded versions of their apps appearing in search results. We analysed the Lumen database [44] and found that the 13 Android markets have sent an [average, stdev, median] of [11.9, 22.2, 0] notices, and been mentioned in [851.4, 937.6, 501]. The 10 iOS markets have sent [0.2, 0.4, 0] and been mentioned in [367.0, 553.9, 54] notices. The full breakdown is included in Table 5.3. This

Table 5.3 DMCA and copyright claims in Lumen made by and which mention each market

Market	Notices sent	Notices mentioned in
Appvn	0	501
RevDL	0	9
HappyMod	42	3 078
MODDROID	1	1 650
APKMODY	73	2 183
androeed	0	0
Rexdl	23	696
5play	0	491
Malavida	0	6
APKDONE	5	1 187
ApkVision	0	659
LMHMOD	11	250
An1	0	358
Mobilism	1	759
Zeus (IPAGalaxy)	0	1
(Malavida)	0	6
OMTK	0	54
AppCake	0	275
PDALIFE	0	691
iOSGods	0	1 801
AppDB	0	34
Platinmods	0	416
TopStore	0	0

gives us an idea of the massive scale of DMCA notices. Furthermore, each notice typically contains multiple offending links from the same and/or different markets.

The markets sending notices is uncommon: seven Android markets send none and only one iOS market sent one. Markets typically sent notices due to other markets hosting their market app or copying content from them. The notices sent by others mainly come from developers or companies representing their interests notifying search engines of the copyright infringement in modded markets. Typically, both types of notices have been sent to Google, the most popular search engine and one of the main contributors to Lumen. However, Google sends copies of the notices to Lumen regardless of the action taken, so it is unknown if they were acted upon. There might be other search engines that have been contacted but did not report to the dataset. Similarly, developer notices reported directly to the markets are not in this database because market operators or developers have not sent copies to Lumen.

5.5 Discussion and possible solutions

This section discusses possible solutions based on our findings.

5.5.1 Technical solutions

Google recently introduced new capabilities in the Android Integrity API in May 2025, including increasing the use of hardware-backed security signals on Android 13, 14 and 15 [94] which run on 59.4% of Android devices as of January 2025 [186]. This Integrity API allows developers to confirm their apps were installed from Google Play and are running the genuine app binaries, and react accordingly. This requires server infrastructure to be effective, otherwise it is trivial to mod. **Apple**, on the other hand, is perceived as doing better by many developers in tackling sideloading, and they also offer an app attestation API for iOS 11 onwards which prevents all non-jailbroken devices from getting a valid attestation, similar to the Android API [20, 21] (99% of users run iOS 16 or newer as of January 2025 [196]). **Developers** should use all the tools available to them. However, these techniques require server-side checks, placing a burden on developers. Furthermore, Google's Integrity API is only recommended in addition to further custom checks as it "works best when used alongside other signals [...] and not as your sole anti-abuse mechanism" [8]. These attestations can also negatively impact genuine users, as mentioned by many developers in our survey; e.g. users with rooted, old or budget devices with limited support for the more reliable attestation techniques.

Apple and Google should increase awareness of available tools, giving a complete overview of all of them in a single place; make template apps and default configurations include obfuscation by default unlike the current defaults [93]; and promote Google Licensing and Integrity APIs. **Google and Android device manufacturers** should also provide longer update support for Android devices, making robust app attestation available to more users (iOS 11 was released in 2017 and Android 13 in 2022). The problem with Android update support goes beyond device manufacturers, as part manufacturers and carriers add to the latency and fragmentation [119, 191, 204]. **Apple** should enable client-side scanning similar to Google Play Protect or give users access to the app package files (IPAs) before and/or after installation, now that third-party markets exist in the EU and users are more aware of sideloading and modded apps. Apple still notarises and signs all apps uploaded to third-party markets in the EU [27], but this obviously does not include the sideloading and modded app markets we have studied.

5.5.2 DMCA copyright notices

More than half the developers do not request takedown through DMCA, demonstrating a lack of faith in the current takedown process as well as a lack of time and resources, especially for smaller developers. Less than 2% had success with DMCA notices. The few that did typically saw their apps reappear in the same markets shortly afterwards. Many would use an **automated DMCA notice** or flagging tool by Google and/or Apple if one existed.

Developers experience a similar situation with **search engine DMCA notices**: by the time search engines such as Google have processed a request to suppress a link from search results, new versions of those apps are already in the modded market which are still visible. We believe the call for an **automated copyright flagging system**, similar to our ModZoo and iOSZoo project pipelines [176], or perhaps a **ban of entire domains** for sites which receive many reports would be a more effective way to protect developers. Since market operators reported that most users arrive at their markets through search engines, a powerful solution to revamp the DMCA notice process could be implemented by search engines. Alternatively, scrapers similar to those we developed to obtain the ModZoo and iOSModZoo datasets could be implemented to automatically flag possible copyright infringing apps. The same is true for **advertisers** which should not serve ads alongside modded apps, since doing so would significantly reduce income for modded app markets. **Google AdSense** is the most popular in our sample, however the wider ecosystem would need to respond if this approach is to be effective in the long term.

Google and Apple as **app market providers** lose their share of commission as a result of free IAPs and pirated paid apps being easily accessible to users on modded app markets. Meanwhile, Google gains ad revenue from the modded markets that use Google AdSense (without having to share this revenue with the original developers). Google has both the most popular search engine, advertising platform and mobile operating system, and helping developers appears to be in their best interest. The incentives for **other search engines and ad networks** are less clear. Therefore, the effectiveness of any attempt to remove modded apps from search results or preventing ads from appearing on modded app markets without **changes in legislation** is less clear.

5.6 Literature review

Goodwin and Woolley survey sideloading forum users and computer science staff and students to explore user motivations and awareness of sideloading: 94% of the forum users and 46% of the academics had sideloaded apps and most had security concerns [92]. The main reason for installing apps from alternative markets was better availability of apps. Some users mentioned markets included in this study. However, they do not study market operator motivations nor the effect modded app markets have on users and developers. Paquet-Clouston analyses private chats between one market operator trying to distribute the Geost botnet (a banking Trojan) in Android apps and their business associates [160]. Economic incentives are the main motivators for the operator and their workers, they operate more than one market and need to keep making new ones due to bans. Our findings confirm many operators control more than one market; and in ModZoo we found many markets share backend infrastructure. Economic incentives are the main motivator for providers of DDoS-for-hire [113]. Similar to our findings, legitimate usage is used to rationalise illegal behaviours.

Others study developers' security practices. Weir et al. survey 330 Android developers, and find that half use no security assurance techniques and less than a quarter have access to security experts [221]. Rauf et al. find freelance developers pay intermittent attention to code security during

code review tasks [171]. Wee et al. survey 37 developers and discover they tend to use their own secure software development frameworks adapted to their needs, instead of readily available ones despite awareness of them [219].

In terms of DMCA notices' effectiveness, Carpou argues the DMCA needs to be amended and automated takedown notices implemented to help ensure infringing content stays down [54]. Penney found through surveys and analysis of blogs and tweets that automated DMCA notices are more likely to result in takedown in the case of blogs and online accounts [162]. The volume of requests is hypothesised to influence providers like Google and Twitter to act.

5.7 Summary

Market operators were primarily motivated by monetary gains, primarily through ads on their market websites. Few perform security testing beyond scanning apps before uploading, only one Android operator had any long-term testing procedure, and many of the iOS operators did not test apps at all. This leads to security concerns for developers and users, as users are enticed by modded features found in these markets and who then install outdated, untested, and illegally modified versions of popular apps.

Our developer survey respondents are Android developers of all sizes whose apps were found in modded markets; more than half are also iOS developers. One in four developers were unaware of modded Android app markets, and two in five were unaware of modded iOS apps. Developers reported revenue losses from IAPs, advertising, sales, third-party API use within their apps, and increased server calls. Developers also experienced reputational damage, increased errors and crashes and additional support requests. Other users of the original versions of the app also experienced negative effects; for example, players of multiplayer games experienced an unfair disadvantage when compared to modded users with access to game cheats. 76% of the developers found mods include premium features for free, 28% found changed advertising settings, including additional ads and changed advertiser IDs in some cases; 12% found removed license and server checks, added permissions (including overlay and accessibility to show ads on top of other apps), and paid apps. This demoralises some developers, and poses security risks for users. Less than half the developers use techniques to combat modding and unauthorised redistribution: some lacked the technical expertise required or found that these techniques did not work or caused problems for legitimate users, and few considered modded apps not a big enough issue. We proposed Apple and Google should increase awareness of the existing tools and Google and Android manufacturers should support and update devices for longer.

The DMCA is not fit for purpose for developers as it is implemented today: only 2% of developers were successful with DMCA takedown notices. Most of the markets studied in Section 5.4 and in the market surveys knowingly host content that infringes on copyright and are not afraid of copyright infringement. These markets use DMCA as a show of compliance and deterrent for developers: some provide no email contact or online form despite stating that they comply with the DMCA and

copyright law. Some even use menacing language on their DMCA complaints pages. Our survey reveals that developers cannot keep track of the large number of third-party markets who host modded versions of their apps. Smaller developers in particular feel helpless and ill-equipped to address the misuse of their apps. Ad revenue provides the majority of funding for modded markets and ad networks appear to do little to prevent the use of ads on these platforms.

Chapter 6

Conclusion and future work

This dissertation presented an overview of the sideloading and modded app markets ecosystem, the incentives to operate these markets, and the effects on modded app and legitimate app users, the original developers, and app markets affected by modded apps. We studied and asked developers about their experience with the technical and legal tools available to them and their effectiveness. Finally, we suggested possible solutions to unauthorised software modifications, both technical and aimed at regulators.

Chapter 2 provided an overview of the history of software distribution and how 'sideloading' was always the most common way of loading computer programmes onto devices (§2.1). This was done by plugging in punch cards in the early days, then tapes, floppy disks, USB drives, CDs, etc. Physical media are now rare, and online distribution is more common. The introduction of official app stores was one of the innovations of smartphones, making them more similar to the game consoles of the time than personal computing devices. Official app stores gave users and developers a reliable central distribution method for apps with security guarantees (§2.2). While Google allowed Android users to sideload apps, iOS was more restrictive, requiring users and developers to abuse developer tools to enable sideloading (§2.3). Apple's practices, often described as 'monopolistic' [76, 77], have earned them scrutiny in the US and regulations in the EU to allow sideloading (§2.4). However, the ecosystem of third-party markets has remained largely unchanged.

We explored the modded apps and sideloading markets ecosystem from multiple fronts. Chapter 3 introduced our methodology and the *ModZoo* dataset, containing over 146 000 modded Android apps, and the analysis of the Android modded app markets and apps ecosystem. Chapter 4 introduced the *iOSModZoo* and *iOSZoo* datasets, containing over 40 000 modded iOS apps and 77 000 related App Store apps, respectively. Chapter 4 also presented an overview of the iOS sideloading and modded app markets ecosystem. Chapters 3 and 4 compared modded apps to their official counterparts and how they affect user privacy and security, what they offer users, and how they affect the original app developers' revenue while generating revenue for the modded market operators and modders. Finally, Chapter 5 presented our modded market operators survey results, highlighting their lack of security checks and their motivations. Chapter 5 also presented our original app developers survey

results, which highlight the different effects developers of all sizes suffer due to modded apps, and their awareness and experiences with existing tools to combat modding and sideloading, including DMCA (Digital Millennium Copyright Act) takedown notices. Finally, Chapter 5 analysed the current DMCA takedown notices process available to copyright holders and its shortcomings, and suggested possible technical and legal solutions to tackle unauthorised software modifications.

6.1 Effects on original app developers, users, and market operators

This section explores our findings on the modded apps ecosystem and the impact of modded apps and markets on the original app developers, users, and official market operators.

Original app developers' potential income is reduced by pirated apps and unlocked premium features in modded versions of their apps. This dissertation presented the results of the first large-scale study of modded Android and iOS apps and their markets, analysing hundreds of thousands of modded apps and their official counterparts from Google Play and the App Store. The vast majority were modified in one or more ways, including those labelled as unmodified. Our results showed that modded app markets provide paid apps for free: around 5% of apps in Android modded markets and almost 30% in iOS modded markets were pirated paid apps and games offered for free. Pirated iOS apps had more than 1 billion downloads in just three markets, worth $12.5 billion. Modded Android apps typically added permissions and ad libraries; 21% had different ad IDs than their Google Play version, suggesting ad revenue may be diverted away from original developers to a third party. However, these markets are unrelated to the genuine developers, and divert or curtail the app purchase, IAPs and ads revenue streams (ads in the market websites and mainly on Android, also in-app ads).

Our developer survey respondents are Android developers of all sizes whose apps were found in modded markets; more than half are also iOS developers. While the majority of original app developers were aware of modding, one in five were unaware of modded Android apps, one in four were unaware of modded Android app markets, and two in five were unaware of modded iOS apps. Developers reported revenue losses from IAPs, advertising, sales, third-party API use within their apps, and increased server calls. Developers also experienced reputational damage, increased errors and crashes and support requests. They reported that 76% of mods include premium features for free, 28% found changed advertising settings, including additional ads and changed advertiser IDs in some cases; 12% found removed license and server checks, added permissions (including overlay and accessibility to show ads on top of other apps), and paid apps. The developer responses suggested this behaviour was demoralising and posed security risks to users: 55% reported revenue losses from in-app purchases, advertising revenue (45%), app sales (39%), and others (13%). Many developers (132, 18.4% of the total) said nothing can be done about modded apps and markets because contacting them for removal is useless (48), it is impossible to keep up with the modded markets (40), or it is particularly difficult for smaller developers (34), etc. 14 developers stated they suffered reputational damage as a result of modded versions of their apps, and three felt demoralised before our study, one of them even felt demotivated to implement new features.

Users benefit from increased competition and choice: modded apps advertise new, desirable features, which our case studies show often, although not always, work. However, users risk their personal data and their privacy being breached by downloading apps whose code has not been verified by any official entity and modified by third-parties. As mentioned above, permissions and ad libraries are typically added to Android modded apps. Although ad-free versions of apps are widely touted, fewer than 3% of Android modded apps and no iOS modded apps had all ads and trackers removed. VirusTotal marked 9% of Android code-modded apps as containing adware, grayware or Trojans, 10 times the rate found in Google Play. These code-modded Android apps often requested additional permissions including dangerous and deprecated ones. Similarly, modded iOS apps were 33 times more likely to be classified as malicious by VirusTotal than their App Store counterparts (1% versus 0.03% in the App Store). Furthermore, installation methods used by most iOS markets directly install apps without giving users access to the files before or after installation. A popular distribution method allows iOS market operators to remotely extract private data from and install apps on users' devices.

The majority of modded apps fell into the gaming category. However, many other popular apps exist on these markets, including modified versions of social media and messaging apps that allow users to hide when they are online and save and screenshot images without notifying other users, and modified versions of Spotify offering ad-free music without subscription. Thus, modded apps also affect leigitimate app users' privacy and security. Malware and spyware might make use of added permissions to access modded users' private information, contacts, etc. Legitimate users also experience negative effects caused by anti-modding measures, such as restricting features and requiring an internet connection, further server checks, etc. Some developers stated these measures negatively affected legitimate users' experience so they had to disable them.

Official app markets experience a reduction in their potential income similar to the developers, mainly from purchase commissions. Allowing sideloading can gain them favour with users and regulators; however, it can also erode developer confidence, as shown by many developers in our survey who had a more favourable view of Apple than Google because of their handling of sideloading.

6.2 Incentives for operating modded app markets

Modded app market operators benefit from the original developers' intellectual property. Our survey of modded app market operators showed many market operators are primarily motivated by monetary gains, mostly obtained through ads on their market websites. Many iOS markets even had paid subscription tiers. Few performed security testing beyond scanning apps before uploading. This leads to security concerns for developers and users, as users are enticed by modded features found in these markets to install outdated, untested, and illegally modified versions of popular apps.

Furthermore, modded Android markets continued to host malicious apps removed from Google Play for a long time. Many Android and iOS modded markets operate as forums where admins and users can upload apps as files for others to download, making it very difficult for these markets to

check the security of all apps they host. Worryingly, our findings point to a steady community of marketplace operators, modders and user base with a growing number of modded apps.

6.3 Possible technical and legal mitigations

All the negative effects summarised above, which affect users, original developers and markets, suggest that further technical and legal solutions or changes to existing ones are necessary to tackle modding effectively. This section summarises our suggested mitigations for OS vendors (Apple, Google and Android device manufacturers), developers, regulators, and lawmakers.

Android offers official support for sideloading, whereas iOS makes it difficult for the average consumer to install apps from outside the official market. Both approaches have positive and negative effects: Android developers and users have better awareness of modded apps and the risks of sideloading, are given tools to analyse apps more easily, but Google lacks controls to stop the sideloading of malicious apps. Google Play Protect flagged only 1 of the 28 modded apps tested in our case study, as well as the 'Apkmody' market app (§3.3.2). Most modded app users simply disable or ignore Play Protect. Meanwhile, Apple's approach gives them pinch points to ban entire markets and modders, and make sideloading difficult. However, iOS users and developers are less aware of sideloading and its popularity, and users do not have security tools at their disposal to analyse sideloaded apps before installation. Furthermore, the less cumbersome the installation method (in iOS), the more trust on the market it requires and the more concerning privacy-wise it is.

Developers should be aware of modded app markets and their practices. Less than 40% of survey respondents were aware of iOS modded apps but more than 80% were aware of Android ones. However, less than half of the developers used techniques to combat modding and unauthorised redistribution: some lacked the technical expertise required or found that these techniques did not work or caused problems for legitimate users; nevertheless, many developers thought modded apps were a significant concern. Developers should use all the tools available to them. However, the most effective techniques require server-side checks, placing a burden on developers, especially smaller and less-technically savvy ones. Furthermore, these attestations can also negatively impact genuine users, as mentioned by many developers in our survey; especially users with rooted, old or budget devices with limited support for the more reliable attestation techniques.

Both Google and Apple should increase developer awareness of their available tools, giving a complete overview of all of them in a single place. Template apps and default configurations should include obfuscation by default unlike current defaults. Google should promote Google Licensing and its new Integrity API capabilities, which now include the use of hardware-backed security signals on Android 13 onwards. Sadly, these Android versions run on only 59.4% of active Android devices (as of January 2025). Apple is perceived to be doing better at tackling sideloading by many developers. In fairness, Apple's policies aimed to make their ecosystem more secure for mobile users have been successful in some ways: sideloading is not as popular as on Android, Apple has added a lot of friction to the process, and they have so far retained control of their platform (all sideloaded apps including

those distributed in the EU must be signed by Apple or an Apple-approved developer or enterprise signature). Stricter control of enterprise signatures and MDM profiles could effectively mitigate the modded apps problem. They also offer an app attestation API for iOS 11 onwards which prevents all non-jailbroken devices from getting a valid attestation, similar to the Android API. However, a substantial 99% of iOS users run iOS 16 or newer (as of January 2025). While these attestations require developers to implement server infrastructure and engineer their apps and games in ways that restrict functionalities depending on these checks to be effective, they are powerful APIs to check apps come from Google Play or the App Store and are running genuine binaries.

Google and Android device manufacturers should follow Apple's lead and provide longer update support for Android devices, making robust app attestation (and security updates) available to more users. In fact, Google has recently announced upcoming limitations for certified Android devices, disallowing sideloading of apps written by unverified developers [88]. Apple, in turn, should follow Google's lead on enabling client-side scanning of apps similar to Google Play Protect, now that third-party markets exist in the EU and users are more aware of sideloading and modded apps. Although Apple notarises and signs all apps uploaded to approved third-party markets in the EU, this obviously does not include sideloading and modded app markets.

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. Possible mitigations range from allowing sideloading while requiring apps to be tested and signed by an approved tester (Apple's current notarisation approach in the EU), to requiring the distribution of alternative market apps through the official market, etc. The purpose of these measures is to add friction to the installation of pirated and unauthorised modified software (apps). These would also offer pinch-points to support regulation and rapid response to security incidents, etc. Ultimately, in order to prevent national censorship and political or company abuse of these policies, users should be given a choice of notarisation provider or the ability to bypass these measures after being informed of the risks. The latter is the option Google will present to users in the near future [86, 88].

In terms of copyright law and DMCA, our findings and developers' survey confirmed that DMCA takedown notices are not fit for purpose as they are implemented today: modded app markets implemented DMCA forms as a show of compliance while knowingly hosting content that infringes on copyright, and less than 2% of developers reported having success with DMCA takedown notices. The few that did typically saw their apps reappear in the same markets shortly afterwards. Less than half of the developers had requested takedown through DMCA; developers lacked faith in the takedown process, they could not keep track of the large number of third-party markets that hosted many modded versions of their apps and lacked the time and resources necessary. Smaller developers in particular felt helpless and ill-equipped to address the misuse of their apps. Another problem is that DMCA is outdated and inadequate for software and apps that rapidly get replaced by newer versions, since DMCA treats different versions and modified copies of the same copyrighted software or app as individual items that need to be reported individually. This is infeasible for developers to report continuously in the hundreds of markets found in this study (or even the dozen most popular ones).

Many developers reported they would use an automated DMCA notice or flagging tool by Google and/or Apple if one existed. Developers can also serve DMCA takedown notices to search engines, but their experience is not much better than with the markets. By the time a search engine such as Google has processed a request to suppress a link from search results, new versions of the apps are already in modded markets and their links still visible in search results in most cases.

We believe that an automated (DMCA) copyright flagging system, similar to our ModZoo and iOSModZoo scraping pipelines, would provide strong protection for developers. Alternatively, a domain-level ban for reoffending sites frequently reported to Google and other search engines, and internet service providers (ISPs) may also prove effective at protecting developers. We leave it to future work to study how users searching the web using chatbots and Large Language Model tools affect this area, and how well takedown notices work on these tools. Ad revenue from modded app markets' websites provides the majority of funding for modded app market operators, yet ad networks appear to do little to prevent the use of ads on these platforms. Ads should not be served alongside (nor inside) modded apps, significantly reducing income for modded app markets (and modders). Google AdSense was the most popular in our datasets, however the wider ecosystem would need to respond if this approach is to be effective in the long term.

6.4 Closing remarks

This work is very timely for users, developers, and market operators (mainly Google and Apple), as well as regulators. Apple is currently facing increased pressure from both EU and US regulators over its closed ecosystem and monopolistic practices. Although Apple has been required to open their ecosystem to third-party app markets in the EU, this has resulted in only a handful of specialised markets appearing. Thus, while Apple is still under scrutiny, their users have not enjoyed the intended more competitive app market in the EU. Furthermore, Apple still charges substantial commissions to third-party markets and developers hosting their apps in them, and each app published in third-party markets must be approved by Apple.

Regulators need to balance competition and fair markets with user security and intellectual property protection. Our results and recommendations are useful for Google, Apple, and app developers who want to protect users from malicious sideloaded and modded apps. Our findings can also inform users of the risks and effects these apps have on their privacy and developer revenue.

References

[1] Yousra Aafer, Wenliang Du, and Heng Yin. DroidAPIMiner: mining API-level features for robust malware detection in Android. In *Proceedings of the International Conference on Security and Privacy in Communication Systems*, pages 86–103. Springer, 2013.

[2] Kevin Allix, Tegawendé F. Bissyandé, Jacques Klein, and Yves Le Traon. Androzoo: collecting millions of android apps for the research community. In *Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16)*, pages 468–471. ACM, 2016.

[3] Fahad Alswaina and Khaled Elleithy. Android malware permission-based multi-class classification using extremely randomized trees. *IEEE Access*, 6:76217–76227, 2018. https://doi.org/10.1109/ACCESS.2018.2883975.

[4] Joseph Amankwah-Amoah. Competing technologies, competing forces: The rise and fall of the floppy disk, 1971–2010. *Technological Forecasting and Social Change*, 107:121–129, 2016.

[5] Ross Anderson. *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons, Ltd, 2020.

[6] Android Developer. App licensing. URL: https://developer.android.com/google/play/licensing, June 2025. Last accessed September 2025.

[7] Android Developer. PackageManager. URL: https://developer.android.com/reference/android/content/pm/PackageManager/#getInstallSourceInfo(java.lang.String), August 2025. Last accessed September 2025.

[8] Android Developer. Play Integrity API. URL: https://developer.android.com/google/play/integrity, January 2025. Last accessed September 2025.

[9] Android Developer. Sign your app. URL: https://developer.android.com/studio/publish/app-signing, January 2025. Last accessed September 2025.

[10] AppDB. Appdb joins investigation against Apple. URL: https://appdb.to/news/539, December 2024. Last accessed September 2025.

[11] AppDB. Apple fights legally. URL: https://appdb.to/news/518, February 2024. Last accessed September 2025.

[12] AppDB. Apple's revenge. URL: https://appdb.to/news/519, February 2024. Last accessed September 2025.

[13] AppDB. An interoperability request has been submitted. URL: https://appdb.to/news/530, May 2024. Last accessed September 2025.

[14] AppDB. Introducing fair in-app purchases and subscriptions. URL: https://appdb.to/news/538, December 2024. Last accessed September 2025.

[15] AppDB. Merry Christmas and Happy New Year! URL: https://appdb.to/news/540, December 2024. Last accessed September 2025.

[16] AppDB. Official app publishing is live! URL: https://appdb.to/news/529, May 2024. Last accessed September 2025.

[17] AppDB. Recent updates and situation with DMA. URL: https://appdb.to/news/533, July 2024. Last accessed September 2025.

[18] AppDB. Appdb PLUS. URL: https://appdb.to/my/plus, 2025. Last accessed September 2025.

[19] Apple Developer. App Store Support. URL: https://developer.apple.com/support/app-store/, 2025. Last accessed September 2025.

[20] Apple Developer. Assessing fraud risk. URL: https://developer.apple.com/documentation/devicecheck/assessing-fraud-risk, 2025. Last accessed September 2025.

[21] Apple Developer. DeviceCheck. URL: https://developer.apple.com/documentation/devicecheck, 2025. Last accessed September 2025.

[22] Apple Developer. Encrypting your app's files. URL: https://developer.apple.com/documentation/uikit/protecting_the_user_s_privacy/encrypting_your_app_s_files, 2025. Last accessed September 2025.

[23] Apple Developer. Getting started as an alternative app marketplace in the European Union. URL: https://developer.apple.com/support/alternative-app-marketplace-in-the-eu/, 2025. Last accessed September 2025.

[24] Apple Developer. Keychain services. URL: https://developer.apple.com/documentation/security/keychain-services, 2025. Last accessed September 2025.

[25] Apple Developer. Requesting interoperability with iOS and iPadOS in the European Union. URL: https://developer.apple.com/support/ios-interoperability/, 2025. Last accessed September 2025.

[26] Apple Developer. Understanding the Core Technology Fee for iOS and iPadOS apps in the European Union. URL: https://developer.apple.com/support/core-technology-fee/, 2025. Last accessed September 2025.

[27] Apple Developer. Update on apps distributed in the European Union. URL: https://developer.apple.com/support/dma-and-apps-in-the-eu/, 2025. Last accessed September 2025.

[28] Apple Developer. Update on apps distributed in the European Union – Core Technology Comission. URL: https://developer.apple.com/support/dma-and-apps-in-the-eu/#core-technology-commission, 2025. Last accessed September 2025.

[29] Apple Inc. Apple unveils higher quality DRM-free music on the iTunes Store. URL: https://web.archive.org/web/20070408155651/http://www.apple.com/pr/library/2007/04/02itunes.html, April 2007. Last accessed September 2025.

[30] Apple Inc. App Store stopped over $7 billion in potentially fraudulent transactions in four years. URL: https://www.apple.com/newsroom/2024/05/app-store-stopped-over-7-billion-usd-in-potentially-fraudulent-transactions/, May 2024. Last accessed September 2025.

[31] Apple Inc. Apple announces changes to iOS, Safari, and the App Store in the European Union. URL: https://www.apple.com/newsroom/2024/01/apple-announces-changes-to-ios-safari-and-the-app-store-in-the-european-union/, January 2024. Last accessed September 2025.

[32] Apple Inc. European Digital Markets Act (DMA). URL: https://www.apple.com/legal/dma/, 2025. Last accessed September 2025.

[33] Apple Support. About App Store security. URL: https://support.apple.com/guide/security/secb8f887a15, December 2024. Last accessed September 2025.

[34] Apple Support. App code signing process in iOS, iPadOS, tvOS, watchOS and visionOS. URL: https://support.apple.com/guide/security/sec7c917bf14, December 2024. Last accessed September 2025.

[35] Apple Support. Secure Enclave. URL: https://support.apple.com/guide/security/sec59b0b31ff, May 2024. Last accessed September 2025.

[36] AM Aswini and P Vinod. Droid permission miner: Mining prominent permissions for Android malware analysis. In *Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, pages 81–86. IEEE, 2014.

[37] Samuel Axon. Apple revokes Facebook's developer certificate over data-snooping app—Google could be next. URL: https://arstechnica.com/gadgets/2019/01/facebook-and-google-offered-gift-cards-for-root-level-access-to-ios-users-data/, January 2019. Last accessed September 2025.

[38] Michael Backes, Sven Bugiel, and Erik Derr. Reliable third-party library detection in Android and its security applications. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, page 356–367. Association for Computing Machinery, 2016.

[39] Richard Baguley. The gadget we miss: The BBC Micro. URL: https://medium.com/people-gadgets/the-gadget-we-miss-the-bbc-micro-cde0cb73bce3, July 2013. Last accessed September 2025.

[40] Monica J Barratt, Simon Lenton, Alexia Maddox, and Matthew Allen. 'what if you live on top of a bakery and you like cakes?'—drug use and harm trajectories before, during and after the emergence of silk road. *International Journal of Drug Policy*, 35:50–57, 2016.

[41] Ryad Benadjila and Mathieu Renard. Security offense and defense strategies: Video-game consoles architecture under microscope. URL: https://www.slideshare.net/slideshow/security-offense-and-defense-strategies-videogame-consoles-architecture-under-microscope/63910924, July 2016. Last accessed September 2025.

[42] Bruce L Berg. *Qualitative research methods for the social sciences*. Allyn & Bacon, 2006.

[43] Andréanne Bergeron, David Décary-Hétu, and Marie Ouellet. Conflict and victimization in online drug markets. *Victims & Offenders*, 17(3):350–371, 2022.

[44] Berkman Klein Center for Internet & Society at Harvard University. Lumen. URL: https://lumendatabase.org, 2017. Last accessed September 2025.

[45] Ron Aquino Bethel Otuteye, Khawaja Shams. How we kept the Google Play & Android app ecosystems safe in 2024. URL: https://security.googleblog.com/2025/01/how-we-kept-google-play-android-app-ecosystem-safe-2024.html, January 2025. Last accessed September 2025.

[46] Ian Birnbaum. The state of PC piracy in 2016. URL: https://www.pcgamer.com/the-state-of-pc-piracy-in-2016/, August 2016. Last accessed November 2025.

[47] Eduardo Blázquez, Sergio Pastrana, Álvaro Feal, Julien Gamba, Platon Kotzias, Narseo Vallina-Rodriguez, and Juan Tapiador. Trouble over-the-air: An analysis of FOTA apps in the Android ecosystem. In *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP)*, pages 1606–1622. IEEE, 2021. https://doi.org/10.1109/SP40001.2021.00095.

[48] C. Scott Brown. Here are the phone update policies from every major Android manufacturer. URL: https://www.androidauthority.com/phone-update-policies-1658633/, November 2025. Last accessed November 2025.

[49] Boštjan Brumen, Aljaž Zajc, and Leon Bošnjak. Permissions vs. privacy policies of apps in Google Play store and Apple App Store. In *Information Modelling and Knowledge Bases XXXIV*, pages 258–275. IOS Press, 2023.

[50] William J Buchanan, Simone Chiale, and Richard Macfarlane. A methodology for the security evaluation within third-party Android marketplaces. *Digital Investigation*, 23:88–98, 2017.

[51] Josh Bycer. The DRM distinction of Steam's success. URL: https://www.gamedeveloper.com/business/the-drm-distinction-of-steam-s-success, May 2013. Last accessed September 2025.

[52] Gregory Campbell. Insert coins to continue: Exploring the realm of freemium games. URL: https://www.gamedeveloper.com/business/insert-coins-to-continue-exploring-the-realm-of-freemium-games, March 2015. Last accessed September 2025.

[53] Carnegie Mellon University. The CMU pronouncing dictionary. Version 0.7b, URL: http://www.speech.cs.cmu.edu/cgi-bin/cmudict, 1993. Last accessed September 2025.

[54] Zoe Carpou. Robots, pirates, and the rise of the automated takedown regime: Using the DMCA to fight piracy and protect end-users. *Colum. JL & Arts*, 39:551, 2015.

[55] Lauren Chadwick. Taking a bite out of Apple: How do its EU regulatory pressures and US antitrust woes compare? URL: https://www.euronews.com/next/2024/03/29/taking-a-bite-out-of-apple-how-do-its-eu-regulatory-pressures-and-us-antitrust-woes-compar, March 2024. Last accessed September 2025.

[56] Brian Chapman. Punched tape codes. URL: https://www.chilton-computing.org.uk/acl/literature/chapman/p015.htm, April 1961. Last accessed September 2025.

[57] Shavina Chau, Joonho Ko, and Jessica Tang. History of hacking the Nintendo 3DS. URL: https://courses.csail.mit.edu/6.857/2019/project/20-Chau-Ko-Tang.pdf, 2019. Last accessed September 2025.

[58] Kai Chen, Xueqiang Wang, Yi Chen, Peng Wang, Yeonjoon Lee, XiaoFeng Wang, Bin Ma, Aohui Wang, Yingjun Zhang, and Wei Zou. Following devil's footprints: Cross-platform analysis of potentially harmful libraries on Android and iOS. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pages 357–376, 2016. https://doi.org/10.1109/SP.2016.29.

[59] Thomas Claburn. Apple has only 30 days to comply with EU DMA rules. URL: https://www.theregister.com/2025/05/29/apple_has_only_about_30/, May 2025. Last accessed September 2025.

[60] Cloudflare Inc. Distributed denial-of-service (DDoS) protection. URL: https://www.cloudflare.com/ddos/, 2025. Last accessed September 2025.

[61] Cloudflare, Inc. Reporting abuse - Cloudflare. URL: https://www.cloudflare.com/trust-hub/reporting-abuse/, November 2025. Last accessed November 2025.

[62] Computer History Museum. Memory & storage: Timeline of computer history. URL: https://www.computerhistory.org/timeline/memory-storage/, 2025. Last accessed September 2025.

[63] Computer History Museum. Software & languages: Timeline of computer history. URL: https://www.computerhistory.org/timeline/software-languages/, 2025. Last accessed September 2025.

[64] Computer History Museum. Software becomes a product. URL: https://www.computerhistory.org/revolution/mainframe-computers/7/172, 2025. Last accessed September 2025.

[65] Computer History Museum. Timeline: The storage engine. URL: https://www.computerhistory.org/storageengine/timeline/, 2025. Last accessed September 2025.

[66] Rodrigo Copetti. PlayStation 3 architecture. URL: https://www.copetti.org/writings/consoles/playstation-3/, October 2021. Last accessed September 2025.

[67] Cyble Inc. Fake ad blockers targeting Android devices through adware campaign. URL: https://cyble.com/blog/fake-ad-blockers-targeting-android-devices-through-adware-campaign/, July 2021. Last accessed November 2025.

[68] Zhui Deng, Brendan Saltaformaggio, Xiangyu Zhang, and Dongyan Xu. iRiS: Vetting private API abuse in iOS applications. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*, pages 44–56. Association for Computing Machinery, 2015.

[69] Norman Desmarais. CD-ROM as a software distribution medium. *Optical information systems*, 9(4):209–13, 1989.

[70] Harvey M Dietel and Barbara Deitel. *An introduction to information processing*. Academic Press, 1986.

[71] Cory Doctorow. New AACS processing key leaks onto the net. URL: https://web.archive.org/web/20090124055806/http://www.boingboing.net/2007/05/30/new-aacs-processing-.html, May 2007. Last accessed September 2025.

[72] Gary Dusek, Yuilya Yurova, and Cynthia P Ruppel. Using social media and targeted snowball sampling to survey a hard-to-reach population: A case study. *International Journal of doctoral studies*, 10:279, 2015.

[73] dwyl.io. List of English words. URL: https://github.com/dwyl/english-words/, 2025. Last accessed September 2025.

[74] David Décary-Hétu and Benoit Dupont. Reputation in a dark network of online criminals. *Global Crime*, 14(2-3):175–196, 2013.

[75] Electronic Frontier Foundation. Sony BMG litigation info. URL: https://www.eff.org/cases/sony-bmg-litigation-info, July 2007. Last accessed September 2025.

[76] European Commission. Commission fines Apple over €1.8 billion over abusive App Store rules for music streaming providers. URL: https://ec.europa.eu/commission/presscorner/detail/en/ip_24_1161, March 2024. Last accessed September 2025.

[77] European Commission. Commission finds Apple and Meta in breach of the Digital Markets Act. URL: https://ec.europa.eu/commission/presscorner/detail/en/ip_25_1085, April 2025. Last accessed September 2025.

[78] European Commission. Commission provides guidance under Digital Markets Act to facilitate development of innovative products on Apple's platforms. URL: https://digital-markets-act.ec.europa.eu/commission-provides-guidance-under-digital-markets-act-facilitate-development-innovative-products-2025-03-19_en, March 2025. Last accessed September 2025.

[79] European Commission. DMA designated gatekeepers. URL: https://digital-markets-act.ec.europa.eu/gatekeepers_en, 2025. Last accessed September 2025.

[80] European Parliament. 2020/0374(COD) Digital Markets Act, 2021.

[81] European Parliament. Deal on Digital Markets Act: EU rules to ensure fair competition and more choice for users. URL: https://www.europarl.europa.eu/news/en/press-room/20220315IPR25504, 2022. Last accessed September 2025.

[82] Chris Evans. Weak bits floppy disc protection: an alternate origins story on 8-bit. URL: https://scarybeastsecurity.blogspot.com/2020/06/weak-bits-floppy-disc-protection.html, June 2020. Last accessed September 2025.

[83] Johannes Feichtner, David Missmann, and Raphael Spreitzer. Automated binary analysis on iOS: A case study on cryptographic misuse in iOS applications. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '18)*, page 236–247. Association for Computing Machinery, 2018.

[84] Michael Felstead. Identifying factors that influence the use of dark web cryptomarkets: Qualitative interviews with cryptomarket users. *The Plymouth Law and Criminal Justice Review*, 10(1):84–104, 2018.

[85] Wes Fenlon. PC piracy survey results: 35 percent of PC gamers pirate. URL: https://www.pcgamer.com/pc-piracy-survey-results-35-percent-of-pc-gamers-pirate/, August 2016. Last accessed November 2025.

[86] Matthew Forsythe. Android developer verification: Early access starts now as we continue to build with your feedback. https://android-developers.googleblog.com/2025/11/android-developer-verification-early.html, November 2025. Last accessed November 2025.

[87] Yanick Fratantonio, Chenxiong Qian, Simon Chung, and Wenke Lee. Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop. In *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*, San Jose, CA, May 2017.

[88] Suzanne Frey. A new layer of security for certified Android devices. https://android-developers.googleblog.com/2025/08/elevating-android-security.html, August 2025. Last accessed November 2025.

[89] David J Garney. A software structure for testing a complex product with a minicomputer. In *Proceedings of the 1971 26th annual conference (ACM '71)*, pages 191–196. Association for Computing Machinery, 1971.

[90] Chaim Gartenberg. Spotify reveals 2 million free users are dodging ads. URL: https://www.theverge.com/2018/3/23/17156014/spotify-users-premium-modded-hacked-app-free-streaming-music, March 2018. Last accessed September 2025.

[91] Eric Goldman and Sebastian Schwemer. How the DMCA anticipated the DSA's due process obligations. URL: https://verfassungsblog.de/how-the-dmca-anticipated-the-dsas-due-process-obligations/, February 2024. Last accessed November 2025.

[92] Craig Goodwin and Sandra Woolley. Sideloading: An exploration of drivers and motivations. In *Proceedings of the 35th British HCI and Doctoral Consortium*, pages 1–6, 2022. http://doi.org/10.14236/ewic/HCI2022.37.

[93] Google for Developers. Enable app optimization. URL: https://developer.android.com/build/shrink-code, July 2025. Last accessed September 2025.

[94] Google for Developers. Improved verdicts in android 13 and later devices. URL: https://developer.android.com/google/play/integrity/improvements, September 2025. Last accessed September 2025.

[95] Google LLC. Changes to Google Play's billing system for users in Russia and Belarus. URL: https://support.google.com/googleplay/android-developer/answer/11950272, March 2022. Last accessed September 2025.

[96] Google LLC. Android 15 compatibility definition. URL: https://source.android.com/docs/compatibility/15/android-15-cdd, November 2025. Last accessed November 2025.

[97] Google LLC. API reference: Manifest.permission. URL: https://developer.android.com/reference/android/Manifest.permission, November 2025. Last accessed November 2025.

[98] Google LLC. Choose a category and tags for your app or game. URL: https://support.google.com/googleplay/answer/9859673, November 2025. Last accessed November 2025.

[99] Google LLC. Content delistings due to copyright - Google Transparency Report. URL: https://transparencyreport.google.com/copyright/overview, November 2025. Last accessed November 2025.

[100] Google LLC. Google Play Protect. URL: https://developers.google.com/android/play-protect, November 2025. Last accessed November 2025.

[101] Google LLC. Learn about refunds on Google Play. URL: https://support.google.com/googleplay/answer/2479637, 2025. Last accessed September 2025.

[102] Google LLC. Rest of the world. URL: https://support.google.com/googleplay/android-developer/answer/12201481, 2025. Last accessed September 2025.

[103] Google LLC. Supported locations for distribution to Google Play users. URL: https://support.google.com/googleplay/android-developer/answer/10532353, January 2025. Last accessed September 2025.

[104] Google LLC. Use Google Play Protect to help keep your apps safe and your data private. URL: https://support.google.com/googleplay/answer/2812853, November 2025. Last accessed November 2025.

[105] P H Grinyer. What were punch cards and how did they change business? URL: https://www.bcs.org/articles-opinion-and-research/what-were-punch-cards-and-how-did-they-change-business/, November 2021. Last accessed September 2025.

[106] Guardsquare. ProGuard. URL: https://www.guardsquare.com/proguard, 2025. Last accessed September 2025.

[107] Mathieu Guillot. Survey of darknet carders. 2019.

[108] Jingyi Guo, Min Zheng, Yajin Zhou, Haoyu Wang, Lei Wu, Xiapu Luo, and Kui Ren. ilibscope: Reliable third-party library detection for ios mobile apps. *arXiv preprint arXiv:2207.01837*, 2022.

[109] Marie Charlotte Götting. Spotify's revenues from 2012 to 2023 by segment. URL: https://www.statista.com/statistics/245125/, February 2025. Last accessed September 2025.

[110] Kaspar Hageman, Álvaro Feal, Julien Gamba, Aniketh Girish, Jakob Bleier, Martina Lindorfer, Juan Tapiador, and Narseo Vallina-Rodriguez. Mixed signals: Analyzing software attribution challenges in the android ecosystem. *IEEE Transactions on Software Engineering*, 2023.

[111] Thomas Haigh. Software in the 1960s as concept, service, and product. *IEEE Annals of the History of Computing*, 24(1):5–13, 2002.

[112] HM Government. Copyright, Designs and Patents Act 1988. URL: https://www.legislation.gov.uk/ukpga/1988/48/, November 2025. Last accessed November 2025.

[113] Alice Hutchings and Richard Clayton. Exploring the provision of online booter services. *Deviant Behavior*, 37(10):1163–1178, 2016.

[114] iBotPeaches. Apktool. URL: https://github.com/iBotPeaches/Apktool, 2025. Last accessed September 2025.

[115] Muhammad Ikram, Narseo Vallina-Rodriguez, Suranga Seneviratne, Mohamed Ali Kaafar, and Vern Paxson. An analysis of the privacy and security risks of Android VPN permission-enabled apps. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, pages 349–364, 2016. https://doi.org/10.1145/2987443.2987471.

[116] Jay Freeman. Cydia installer. URL: https://git.saurik.com/cydia.git, May 2019. Last accessed November 2025.

[117] Peter Jennings. Microchess for the Kim-1. URL: https://benlo.com/microchess/index.html, March 2006. Last accessed September 2025.

[118] JetBrains. Mobile operating systems targeted by developers worldwide in 2022 and 2023 [graph]. URL: https://www.statista.com/statistics/1078678/software-development-operating-system-mobile/, November 2023. Last accessed September 2025.

[119] Kailani R Jones, Ting-Fang Yen, Sathya Chandran Sundaramurthy, and Alexandru G Bardas. Deploying Android security updates: an extensive study involving manufacturers, carriers, and end users. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 551–567, 2020.

[120] Alex Kantchelian, Michael Carl Tschantz, Sadia Afroz, Brad Miller, Vaishaal Shankar, Rekha Bachwani, Anthony D Joseph, and J Doug Tygar. Better malware ground truth: Techniques for weighting anti-virus vendor labels. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security (AISec '15)*, pages 45–56, 2015. https://doi.org/10.1145/2808769.2808780.

[121] Gregg Keizer. Microsoft kills Office anti-piracy program. URL: https://www.computerworld.com/article/1459527/microsoft-kills-office-anti-piracy-program-2.html, December 2010. Last accessed September 2025.

[122] Andy Kelly. Code wheels, poison, and star maps: the creative ways old games fought piracy. URL: https://www.pcgamer.com/code-wheels-poison-and-star-maps-the-creative-ways-old-games-fought-piracy/, August 2020. Last accessed September 2025.

[123] Dara Kerr, Nick Robins-Early, and Johana Bhuiyan. 'Slap on the wrist': critics decry weak penalties on Google after landmark monopoly trial. URL: https://www.theguardian.com/technology/2025/sep/03/google-monopoly-case-ruling, September 2025. Last accessed September 2025.

[124] Kobra Khanmohammadi, Neda Ebrahimi, Abdelwahab Hamou-Lhadj, and Raphaël Khoury. Empirical study of Android repackaged applications. *Empirical Software Engineering*, 24:3587–3629, 2019.

[125] Kim Jong Cracks. Clutch. URL: https://github.com/KJCracks/Clutch, 2024. Last accessed September 2025.

[126] Aaron Klotz. Denuvo claims its DRM does not hinder gaming performance. URL: https://www.tomshardware.com/news/denuvo-claims-drm-does-not-hinder-gaming-performance, July 2023. Last accessed September 2025.

[127] Shawn Knight. Capcom strips Denuvo from Resident Evil Village after nearly two years. URL: https://www.techspot.com/news/98259-capcom-strips-denuvo-resident-evil-village-after-nearly.html, April 2023. Last accessed September 2025.

[128] John Koetsier. App developers losing $3-4 billion annually thanks to 14 billion pirated apps. URL: https://www.forbes.com/sites/johnkoetsier/2017/07/24/app-developers-losing-3-4-billion-annually-thanks-to-14-billion-pirated-apps/, July 2017. Last accessed November 2025.

[129] John Koetsier. The mobile economy has a $17.5B leak: App piracy. URL: https://www.forbes.com/sites/johnkoetsier/2018/02/02/app-publishers-lost-17-5b-to-piracy-in-the-last-5-years-says-tapcore/, February 2018. Last accessed November 2025.

[130] John Koetsier. TikTok earned $205 million more than Facebook, Twitter, snap and Instagram combined on in-app purchases in 2023. URL: 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/, March 2023. Last accessed September 2025.

[131] Renuka Kumar, Apurva Virkud, Ram Sundara Raman, Atul Prakash, and Roya Ensafi. A large-scale investigation into geodifferences in mobile apps. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1203–1220. USENIX Association, 2022.

[132] Jedrzej Kunat. What exactly is DRM in video games and why should you care? URL: https://www.gog.com/blog/what-exactly-is-drm-in-video-games-and-why-should-you-care/, November 2024. Last accessed September 2025.

[133] Mark A. Lantz, Simeon Furrer, Martin Petermann, Hugo Rothuizen, Stella Brach, Luzius Kronig, Ilias Iliadis, Beat Weiss, Ed R. Childers, and David Pease. Magnetic tape storage technology. *ACM Trans. Storage*, 21(1), January 2025.

[134] Pierre Laperdrix, Naif Mehanna, Antonin Durey, and Walter 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 (WWW '22)*, pages 3440–3449. Association for Computing Machinery, 2022.

[135] Rimantas Leonavičius. How to access Google Play app store while in China. URL: https://cybernews.com/resources/how-to-access-google-play-app-store-while-in-china/, September 2021. Last accessed September 2025.

[136] Xiang Li, Jianyi Liu, Yanyu Huo, Ru Zhang, and Yuangang Yao. An Android malware detection method based on AndroidManifest file. In *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 239–243. IEEE, 2016. https://doi.org/10.1109/CCIS.2016.7790261.

[137] Licel. DexProtector. URL: https://licelus.com/products/dexprotector, 2025. Last accessed September 2025.

[138] Dexin Liu, Yue Xiao, Chaoqi Zhang, Kaitao Xie, Xiaolong Bai, Shikun Zhang, and Luyi Xing. iHunter: Hunting privacy violations at scale in the software supply chain on iOS. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 5663–5680. USENIX Association, August 2024.

[139] Stanley Miller Longwill, Jesse M McNellis, and Douglas R Posson. The use of minicomputers in a distributed information processing system; feasibility study. Technical report, US Geological Survey, 1980.

[140] Joshua Lund. By default, Signal doesn't Recall. URL: https://signal.org/blog/signal-doesnt-recall/, May 2025. Last accessed September 2025.

[141] Ziang Ma, Haoyu Wang, Yao Guo, and Xiangqun Chen. LibRadar: Fast and accurate detection of third-party libraries in Android apps. In *Proceedings of the 38th International Conference on Software Engineering Companion (ICSE '16)*, page 653–656. Association for Computing Machinery, 2016. https://doi.org/10.1145/2889160.2889178.

[142] Sam Machkovech. Evidence continues to mount about how bad Denuvo is for PC gaming performance. URL: https://arstechnica.com/gaming/2018/12/evidence-continues-to-mount-about-how-bad-denuvo-is-for-pc-gaming-performance/, December 2018. Last accessed September 2025.

[143] Alexia Maddox, Monica J Barratt, Matthew Allen, and Simon Lenton. Constructive activism in the dark web: cryptomarkets and illicit drugs in the digital 'demimonde'. *Information, Communication & Society*, 19(1):111–126, 2016.

[144] Eva-Maria Maier, Leonie Maria Tanczer, and Lukas Daniel Klausner. Surveillance disguised as protection: A comparative analysis of sideloaded and in-store parental control apps. In *Proceedings on Privacy Enhancing Technologies*, 2025.

[145] Everton Da S Maldonado, Rabe Abdalkareem, Emad Shihab, and Alexander Serebrenik. An empirical study on the removal of self-admitted technical debt. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 238–248. IEEE, 2017.

[146] James Martin, Rasmus Munksgaard, Ross Coomber, Jakob Demant, and Monica J Barratt. Selling drugs on darkweb cryptomarkets: differentiated pathways, risks and rewards. *The British Journal of Criminology*, 60(3):559–578, 2020.

[147] Alfonso Maruccia. Developer delves into Denuvo DRM to run Hogwarts Legacy on a secondary PC. URL: https://www.techspot.com/news/102488-developer-delves-denuvo-drm-run-hogwarts-legacy-secondary.html, April 2024. Last accessed September 2025.

[148] Alfonso Maruccia. Steam's DRM was inspired by an exec's nephew and his trusty CD burner. URL: https://www.techspot.com/news/107288-steam-drm-exists-thanks-nephew-trusty-cd-burner.html, March 2025. Last accessed September 2025.

[149] Kimberley Masson and Angus Bancroft. 'nice people doing shady things': Drugs and the morality of exchange in the darknet cryptomarkets. *International Journal of Drug Policy*, 58:78–84, 2018.

[150] Microsoft Source. Nokia and Microsoft sign definitive agreement ahead of schedule. URL: https://news.microsoft.com/source/2011/04/21/nokia-and-microsoft-sign-definitive-agreement-ahead-of-schedule/, April 2011. Last accessed September 2025.

[151] James Middleton. EMI, Apple get rid of DRM. URL: https://www.telecoms.com/mobile-devices/emi-apple-get-rid-of-drm, April 2007. Last accessed September 2025.

[152] George A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, November 1995.

[153] Collins W. Munyendo, Kentrell Owens, Faith Strong, Shaoqi Wang, Adam J. Aviv, Tadayoshi Kohno, and Franziska Roesner. "You have to ignore the dangers": User perceptions of the security and privacy benefits of WhatsApp mods. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 4515–4533. IEEE, 2025.

[154] Calen Nakash. Denuvo DRM proven to hurt performance of games it's attached to. URL: https://www.thegamer.com/denuvo-drm-hurts-game-performance/, March 2019. Last accessed September 2025.

[155] W Lawrence Neuman. *Social research methods, qualitative and quantitative approaches*. Pearson International Edition, 2005.

[156] Office of Public Affairs. Department of Justice wins significant remedies against Google. URL: https://www.justice.gov/opa/pr/department-justice-wins-significant-remedies-against-google, September 2025. Last accessed September 2025.

[157] Damilola Orikogbo, Matthias Büchler, and Manuel Egele. CRiOS: Toward large-scale iOS application analysis. In *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '16)*, page 33–42. Association for Computing Machinery, 2016.

[158] Andrew Orlowski. Why Symbian failed: developers, developers, developers. URL: https://www.theregister.com/2010/11/09/symbian_developers_mailbag/, November 2010. Last accessed September 2025.

[159] Kentrell Owens. *Technology and Power: Examining Imbalances Through Usable Security & Privacy Research*. PhD thesis, University of Washington, 2025.

[160] Masarah-Cynthia Paquet-Clouston. *The Role of Informal Workers in Online Economic Crime*. PhD thesis, Arts & Social Sciences: School of Criminology, 2021.

[161] Peng Peng, Limin Yang, Linhai Song, and Gang Wang. Opening the blackbox of VirusTotal: Analyzing online phishing scan engines. In *Proceedings of the Internet Measurement Conference (IMC '19)*, page 478–485. Association for Computing Machinery, 2019. https://doi.org/10.1145/3355369.3355585.

[162] Jonathon W Penney. Privacy and legal automation: the DMCA as a case study. *Stan. Tech. L. Rev.*, 22:412, 2019.

[163] Sarah Perez. Apple updates the rules for its EU App Store by adding more complicated fees. URL: https://techcrunch.com/2025/06/26/apple-updates-the-rules-for-its-eu-app-store-by-adding-more-complicated-fees/, June 2025. Last accessed September 2025.

[164] David Pierce. Microsoft buying Nokia's phone business in a $7.2 billion bid for its mobile future. URL: https://www.theverge.com/2013/9/2/4688530/microsoft-buys-nokias-devices-and-services-unit-unites-windows-phone, September 2013. Last accessed September 2025.

[165] Pinsent Masons LLP. Senior judge says the UK needs a new Copyright Act. URL: https://www.pinsentmasons.com/out-law/news/senior-judge-says-the-uk-needs-a-new-copyright-act, April 2015. Last accessed November 2025.

[166] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. *arXiv preprint arXiv:1806.01156*, 2018.

[167] Joost Poort, João Quintais, Martin A van der Ende, Anastasia Yagafarova, and Mathijs Hageraats. Global online piracy study. *Amsterdam Law School Research Paper*, (2018-21), 2018. https://doi.org/10.2139/ssrn.3224323.

[168] Liam Proven. Why did Symbian lose out to bigger, slower, less-efficient OSes? URL: https://liam-on-linux.livejournal.com/34761.html, May 2013. Last accessed September 2025.

[169] João Pedro Quintais and Sebastian Felix Schwemer. The interplay between the digital services act and sector regulation: How special is copyright? *European Journal of Risk Regulation*, 13(2):191–217, 2022.

[170] Mishaal Rahman. Android 15 cracks down on sideloaded apps even harder to protect users. URL: https://www.androidauthority.com/android-15-restricted-settings-sideloading-3481098/, September 2024. Last accessed November 2025.

[171] Irum Rauf, Tamara Lopez, Helen Sharp, Marian Petre, Thein Tun, Mark Levine, John Towse, Dirk van der Linden, Awais Rashid, and Bashar Nuseibeh. Influences of developers' perspectives on their engagement with security in code. In *Proceedings of the 15th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE '22)*, page 86–95. Association for Computing Machinery, 2022.

[172] Anna Rawlings, Douglas Peden, and Robert Guthrie. Copyright in United Kingdom. URL: https://www.lexology.com/library/detail.aspx?g=9dfc5e21-91c0-494e-986a-0b1943aa81de, July 2019. Last accessed November 2025.

[173] RedOrbit. The history of mobile phone technology. URL: https://www.redorbit.com/reference/the-history-of-mobile-phone-technology/, July 2014. Last accessed September 2025.

[174] Rafael Rob and Joel 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*, 49(1):29–62, 2006. https://doi.org/10.3386/w10874.

[175] Jeremy Ruston. The home computer that roared: How the BBC Micro shaped our world. URL: https://www.infoq.com/presentations/bbc-micro/, August 2024. Last accessed September 2025.

[176] Luis A. Saavedra, Hridoy S. Dutta, Alastair R. Beresford, and Alice Hutchings. ModZoo: A large-scale study of modded Android apps and their markets. In *Proceedings of The 2024 APWG Symposium on Electronic Crime Research (eCrime)*, pages 162–174. IEEE, 2024.

[177] Luis A. Saavedra, Hridoy S. Dutta, Alastair R. Beresford, and Alice Hutchings. App-solutely modded: Surveying modded app market operators and original app developers. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS '25)*, page 739–758. Association for Computing Machinery, 2025.

[178] Aleieldin 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 (CODASPY '21)*, page 269–280. Association for Computing Machinery, 2021.

[179] Aleieldin Salem, Sebastian Banescu, and Alexander Pretschner. Maat: Automatically analyzing VirusTotal for accurate labeling and effective malware detection. *ACM Transactions on Privacy and Security (TOPS)*, 24(4), July 2021.

[180] Bruce Schneier. Sony secretly installs rootkit on computers. URL: https://www.schneier.com/blog/archives/2005/11/sony_secretly_i_1.html, November 2005. Last accessed September 2025.

[181] Bruce Schneier. Sony's DRM rootkit: The real story. URL: https://www.schneier.com/blog/archives/2005/11/sonys_drm_rootk.html, November 2005. Last accessed September 2025.

[182] Ben Schoon. Android 13 will block sideloaded apps from using accessibility features. URL: https://9to5google.com/2022/05/06/android-13-sideloaded-accessibility/, May 2022. Last accessed November 2025.

[183] Gian Luca Scoccia, Marco Autili, Giovanni Stilo, and Paola Inverardi. An empirical study of privacy labels on the Apple iOS mobile App Store. In *Proceedings of the 9th IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft '22)*, page 114–124. Association for Computing Machinery, 2022.

[184] Marcos Sebastián, Richard Rivera, Platon Kotzias, and Juan Caballero. AVclass: A tool for massive malware labeling. In *Proceedings of the 19th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID 2016)*, pages 230–253. Springer International Publishing, 2016.

[185] Yun Shen, Pierre-Antoine Vervier, and Gianluca Stringhini. A large-scale temporal measurement of Android malicious apps: Persistence, migration, and lessons learned. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1167–1184, 2022.

[186] Ahmed Sherif. Mobile Android version market share worldwide 2018–2025. URL: https://www.statista.com/statistics/921152/mobile-android-version-share-worldwide/, January 2025. Last accessed September 2025.

[187] David Silverman. *Doing qualitative research: a practical handbook*. SAGE Publications, 1999.

[188] Michael D Smith and Rahul Telang. Assessing the academic literature regarding the impact of media piracy on sales. *SSRN Electronic Journal*, 2012. https://dx.doi.org/10.2139/ssrn.2132153.

[189] Kristin Snyder. The secret to TikTok's success with in-app purchases. URL: https://dot.la/tiktok-revenue-2659494404.html, March 2023. Last accessed September 2025.

[190] Karina Sokolova, Charles Perez, and Marc Lemercier. Android application classification and anomaly detection with graph-based permission patterns. *Decision Support Systems*, 93:62–76, 2017.

[191] Chad Spensky, Jeffrey Stewart, Arkady Yerukhimovich, Richard Shay, Ari Trachtenberg, Rick Housley, and Robert K Cunningham. Sok: Privacy on mobile devices–it's complicated. *Proceedings on Privacy Enhancing Technologies (PoPETS 2016)*, 2016.

[192] Lukas Stefanko. Turn the light on and give me your passwords! URL: https://www.welivesecurity.com/2017/04/19/turn-light-give-passwords/, April 2017. Last accessed November 2025.

[193] Guillermo Suarez-Tangil, Santanu Kumar Dash, Mansour Ahmadi, Johannes Kinder, Giorgio Giacinto, and Lorenzo 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 (CODASPY '17)*, page 309–320. Association for Computing Machinery, 2017.

[194] Altyeb Altaher Taha and Sharaf Jameel Malebary. Hybrid classification of Android malware based on fuzzy clustering and the gradient boosting machine. *Neural Computing and Applications*, 33(12):6721–6732, 2021.

[195] Zhushou Tang, Ke Tang, Minhui Xue, Yuan Tian, Sen Chen, Muhammad Ikram, Tielei Wang, and Haojin Zhu. iOS, your OS, everybody's OS: Vetting and analyzing network services of iOS applications. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security 20)*, pages 2415–2432, 2020.

[196] TelemetryDeck. iOS versions market share in 2025. URL: https://telemetrydeck.com/survey/apple/iOS/majorSystemVersions/, September 2025. Last accessed September 2025.

[197] Riley Testut. Thoughts on the App Store. URL: https://rileytestut.com/blog/2020/10/14/thoughts-on-app-store/, 2020. Last accessed September 2025.

[198] Riley Testut. Introducing AltStore PAL. URL: https://rileytestut.com/blog/2024/04/17/introducing-altstore-pal/, April 2024. Last accessed September 2025.

[199] The Apple Wiki. Jailbreak. URL: https://theapplewiki.com/wiki/Jailbreak, November 2025. Last accessed November 2025.

[200] The European Parliament and the Council of the European Union. Directive - 2000/31 - EN - e-commerce directive - EUR-Lex. URL: http://data.europa.eu/eli/dir/2000/31/oj, June 2000. Last accessed November 2025.

[201] The European Parliament and the Council of the European Union. Regulation - 2022/2065 - EN - DSA - EUR-Lex. URL: https://eur-lex.europa.eu/eli/reg/2022/2065/oj, October 2022. Last accessed November 2025.

[202] The Netherlands Authority for Consumers & Markets. Market study into mobile app stores. Case no.: ACM/18/032693. URL: https://www.acm.nl/sites/default/files/documents/market-study-into-mobile-app-stores.pdf, April 2019. Last accessed September 2025.

[203] The UK Competition and Markets Authority (CMA). Mobile ecosystems market study final report. URL: https://www.gov.uk/government/publications/mobile-ecosystems-market-study-final-report, August 2022. Last accessed September 2025.

[204] Daniel R Thomas, Alastair R Beresford, and Andrew Rice. Security metrics for the Android ecosystem. In *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 87–98, 2015.

[205] Rob Thubron. All games impacted by the Alder Lake/DRM issues have now been fixed. URL: https://www.techspot.com/news/92942-all-games-impacted-alder-lakedrm-issues-have-now.html, January 2022. Last accessed September 2025.

[206] Daniil Tiganov, Jeff Cho, Karim Ali, and Julian Dolby. SWAN: A static analysis framework for Swift. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020)*, page 1640–1644. Association for Computing Machinery, 2020.

[207] Bao Tran. International copyright laws vs. DMCA what platforms must know. URL: https://patentpc.com/blog/international-copyright-laws-vs-dmca-what-platforms-must-know, November 2025. Last accessed November 2025.

[208] Ash Turner. How many apps are there in the world (2025). URL: https://www.bankmycell.com/blog/number-of-mobile-apps-worldwide, November 2025. Last accessed November 2025.

[209] United States District Court for the Northern District of California. Epic games, inc. v. apple inc. No. 4:20-cv-05640-YGR, Document 1508 (N.D. Cal.), 2021.

[210] U.S. Department of Justice. Justice Department sues Apple for monopolizing smartphone markets. URL: https://www.justice.gov/archives/opa/pr/justice-department-sues-apple-monopolizing-smartphone-markets, 2025. Last accessed October 2025.

[211] Marie Claire Van Hout and Tim Bingham. Responsible vendors, intelligent consumers: Silk Road, the online revolution in drug trading. *International Journal of Drug Policy*, 25(2):183–189, 2014.

[212] Robert Wahl. Cybersecurity: Implications of side-loading applications on mobile devices. In *Proceedings of the International Conference on Life Sciences, Engineering and Technology (ILSET) 2024*, pages 119–125. ISTES, 2024.

[213] Wallhax. What are ESP cheats? how ESP hacks work in multiplayer games! URL: https://wallhax.com/what-are-esp-cheats/, August 2021. Last accessed September 2025.

[214] Haoyu Wang, Zhe Liu, Jingyue Liang, Narseo Vallina-Rodriguez, Yao Guo, Li Li, Juan Tapiador, Jingcun Cao, and Guoai Xu. Beyond Google Play: A large-scale comparative study of Chinese Android app markets. In *Proceedings of the Internet Measurement Conference 2018 (IMC '18)*, pages 293–307. Association for Computing Machinery, 2018. https://doi.org/10.1145/3278532.3278558.

[215] Haoyu Wang, Junjun Si, Hao Li, and Yao Guo. RmvDroid: Towards a reliable Android malware dataset with app metadata. In *Proceedings of the IEEE/ACM 16th International Conference on Mining Software Repositories (MSR 2019)*, pages 404–408, 2019. https://doi.org/10.1109/MSR.2019.00067.

[216] Qing Wang, Juanru Li, Yuanyuan Zhang, Hui Wang, Yikun Hu, Bodong Li, and Dawu Gu. NativeSpeaker: Identifying crypto misuses in Android native code libraries. In *International Conference on Information Security and Cryptology*, pages 301–320. Springer, 2018.

[217] Yan Wang, Haowei Wu, Hailong Zhang, and Atanas Rountev. Orlis: Obfuscation-resilient library detection for Android. In *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft '18)*, pages 13–23. Association for Computing Machinery, 2018.

[218] Jess Weatherbed. The first 'approved' iPhone porn app is coming to Europe. URL: https://www.theverge.com/news/604937/iphone-ios-porn-app-hot-tub-altstore-pal-eu, February 2025. Last accessed September 2025.

[219] Alex Wee, Arina Kudriavtseva, and Olga Gadyatskaya. "Have heard of it": A study with practitioners on adoption of secure software development frameworks. In *Proceedings of the 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 626–633, 2024.

[220] Tao Wei, Min Zheng, Hui Xue, and Dawn Song. Apple without a shell–iOS under targeted attack. In *Proceedings of the Virus Bulletin Conference*, 2014.

[221] Charles Weir, Ben Hermann, and Sascha Fahl. From needs to actions to secure apps? the effect of requirements and developer practices on app security. In *Proceedings of the 29th USENIX security symposium (USENIX security 20)*, pages 289–305, 2020.

[222] Wikibooks. PSP/Homebrew history. URL: https://en.wikibooks.org/wiki/PSP/Homebrew_History, August 2023. Last accessed September 2025.

[223] Jiexin Zhang, Alastair R. Beresford, and Stephan 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 (ISSTA 2019)*, pages 55–65. Association for Computing Machinery, 2019.

[224] Min Zheng, Hui Xue, Yulong Zhang, Tao Wei, and John CS Lui. Enpublic apps: Security threats using iOS enterprise and developer certificates. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 463–474. Association for Computing Machinery, 2015.

[225] Shuofei Zhu, Ziyi Zhang, Limin Yang, Linhai Song, and Gang Wang. Benchmarking label dynamics of VirusTotal engines. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*, pages 2081–2083. Association for Computing Machinery, 2020.

Appendix A

Modded app markets

A.1 Modded app market identification

A.1.1 List of Android keywords

The full list of keywords used to identify the Android markets is presented in Table A.1.

Table A.1 List of keywords used to identify the Android markets.

Keyword	Language
Android app stores	English/Chinese/Hindi/Russian
free Android app store	English/Chinese/Hindi/Russian
mod APK	English/Chinese/Hindi/Russian
download premium APK	English/Chinese/Hindi/Russian
download paid apps free	English/Chinese/Hindi/Russian
mod Android	English/Chinese/Hindi/Russian
mod games	English/Chinese/Hindi/Russian
paid apps for free	English/Chinese/Hindi/Russian
premium apps for free	English/Chinese/Hindi/Russian
unlocked Android apps	English/Chinese/Hindi/Russian
unlocked Android games	English/Chinese/Hindi/Russian
mod apps for free	English/Chinese/Hindi/Russian
YouTube mod	English
Spotify mod	English
Truecaller mod	English

A.1.2 List of iOS keywords

The full list of keywords used to identify the iOS markets is presented in Table A.2.

Table A.2 List of keywords used to identify the iOS markets.

Keyword	Language
iOS app stores	English/Chinese/Hindi/Russian
free iOS app store	English/Chinese/Hindi/Russian
mod IPA	English/Chinese/Hindi/Russian
download premium IPA	English/Chinese/Hindi/Russian
download paid apps free	English/Chinese/Hindi/Russian
mod iOS	English/Chinese/Hindi/Russian
mod games	English/Chinese/Hindi/Russian
paid apps for free	English/Chinese/Hindi/Russian
premium apps for free	English/Chinese/Hindi/Russian
unlocked iOS apps	English/Chinese/Hindi/Russian
unlocked iOS games	English/Chinese/Hindi/Russian
mod apps for free	English/Chinese/Hindi/Russian
YouTube mod	English
Spotify mod	English
Truecaller mod	English

A.2 List of ranked iOS markets

The ranked list of iOS sideloading and modded app markets is presented in Table A.3, it includes the relative rank of each scraped market, and the raw rank among all markets. The reasons why some markets were not scraped are explained in Section 4.2.2.

A.3 Most common third-party hosting services

Table A.4 shows the most common third-party hosting services used by the AppDB, Platinmods and AppCake iOS modded app markets, and the number of apps each market hosts in each.

A.4 List of iOS permissions used in the analysis

The full list of iOS permissions used to identify permissions declared in each app's `Info.plist` file is presented in Table A.5.

Table A.3 List of ranked iOS markets

Rank scraped	Rank	Market
1	1	Mobilism
2	2	Platinmods
N/A	3	AltStore
3	4	PDALife
N/A	5	Scarlet
N/A	6	Panda Helper
4	7	iOSGods
5	8	AppCake
6	9	AppDB
N/A	10	TutuApp
7	11	IPAOMTK
8	12	Malavida
9	13	TopStore
10	14	Zeus

Table A.4 Most common third-party hosting services used across modded iOS app markets.

Modded app market	**Third-party hosting domain**	**Count**
AppDB	sendspace.com	10033
AppDB	mega.nz	6055
AppDB	hostigram.xyz	3793
AppDB	mega.co.nz	3195
AppDB	www.mediafire.com	1379
AppDB	starfiles.co	262
AppDB	github.com	220
Platinmods	drive.google.com	4
Platinmods	apkadmin.com	1
AppCake	userscloud.com	3
AppCake	archive.org	1
AppCake	drive.google.com	1
AppCake	mega.nz	1

Table A.5 List of iOS permissions.

Permission Name	Description
NSBluetoothAlwaysUsageDescription	Privacy - Bluetooth Always Usage Description
NSCalendarsFullAccessUsageDescription	Privacy - Calendars Full Access Usage Description
NSCalendarsWriteOnlyAccessUsageDescription	Privacy - Calendars Write Only Usage Description
NSRemindersFullAccessUsageDescription	Privacy - Reminders Full Access Usage Description
NSCameraUsageDescription	Privacy - Camera Usage Description
NSMicrophoneUsageDescription	Privacy - Microphone Usage Description
NSContactsUsageDescription	Privacy - Contacts Usage Description
NSFaceIDUsageDescription	Privacy - Face ID Usage Description
NSDesktopFolderUsageDescription	Privacy - Desktop Folder Usage Description
NSDocumentsFolderUsageDescription	Privacy - Documents Folder Usage Description
NSDownloadsFolderUsageDescription	Privacy - Downloads Folder Usage Description
NSNetworkVolumesUsageDescription	Privacy - Network Volumes Usage Description
NSRemovableVolumesUsageDescription	Privacy - Removable Volumes Usage Description
NSFileProviderDomainUsageDescription	Privacy - Access to a File Provider Domain Usage Description
NSGKFriendListUsageDescription	Privacy - GameKit Friend List Usage Description
NSHealthClinicalHealthRecordsShareUsageDescription	Privacy - Health Records Usage Description
NSHealthShareUsageDescription	Privacy - Health Share Usage Description
NSHealthUpdateUsageDescription	Privacy - Health Update Usage Description
NSHealthRequiredReadAuthorizationTypeIdentifiers	
NSHomeKitUsageDescription	Privacy - HomeKit Usage Description
NSLocationAlwaysAndWhenInUseUsageDescription	Privacy - Location Always and When In Use Usage Description
NSLocationUsageDescription	Privacy - Location Usage Description
NSLocationWhenInUseUsageDescription	Privacy - Location When In Use Usage Description
NSLocationTemporaryUsageDescriptionDictionary	Privacy - Location Temporary Usage Description Dictionary

NSWidgetWantsLocation	Widget wants location
NSLocationDefaultAccuracyReduced	Privacy - Location Default Accuracy Reduced
NSAppleMusicUsageDescription	Privacy - Media Library Usage Description
NSMotionUsageDescription	Privacy - Motion Usage Description
NSFallDetectionUsageDescription	Fall Detection Usage Description
NSLocalNetworkUsageDescription	Privacy - Local Network Usage Description
NSNearbyInteractionUsageDescription	Privacy - Nearby Interaction Usage Description
NFCReaderUsageDescription	Privacy - NFC Scan Usage Description
NSPhotoLibraryAddUsageDescription	Privacy - Photo Library Additions Usage Description
NSPhotoLibraryUsageDescription	Privacy - Photo Library Usage Description
NSAppleScriptEnabled	Scriptable
NSUpdateSecurityPolicy	
NSAppDataUsageDescription	Privacy - Other Application Data Usage Description
NSUserTrackingUsageDescription	Privacy - Tracking Usage Description
NSAppleEventsUsageDescription	Privacy - AppleEvents Sending Usage Description
NSSystemAdministrationUsageDescription	Privacy - System Administration Usage Description
ITSAppUsesNonExemptEncryption	App Uses Non-Exempt Encryption
ITSEncryptionExportComplianceCode	App Encryption Export Compliance Code
NSSensorKitUsageDescription	Privacy - SensorKit Usage Description
NSSensorKitUsageDetail	
NSSensorKitPrivacyPolicyURL	Privacy - SensorKit Privacy Policy URL
NSSiriUsageDescription	Privacy - Siri Usage Description
NSSpeechRecognitionUsageDescription	Privacy - Speech Recognition Usage Description
NSVideoSubscriberAccountUsageDescription	Privacy - Video Subscriber Account Usage Description
NSWorldSensingUsageDescription	
NSHandsTrackingUsageDescription	

NSIdentityUsageDescription	Privacy - Identity Usage Description
UIRequiresPersistentWiFi	Application uses Wi-Fi
NSBluetoothPeripheralUsageDescription	Privacy - Bluetooth Peripheral Usage Description
NSLocationAlwaysUsageDescription	Privacy - Location Always Usage Description
NSNearbyInteractionAllowOnceUsageDescription	Privacy - Nearby Interaction Allow Once Usage Description
NSCalendarsUsageDescription	Privacy - Calendars Usage Description
NSRemindersUsageDescription	Privacy - Reminders Usage Description
NSBluetoothWhileInUseUsageDescription	Privacy - Bluetooth While In Use Usage Description
OSBundleUsageDescription	Privacy - Driver Extension Usage Description
NSFileProviderPresenceUsageDescription	Privacy - File Provider Presence Usage Description
NSFocusStatusUsageDescription	Privacy - Focus Status Usage Description
NSSystemExtensionUsageDescription	Privacy - System Extension Usage Description
NSVideoSubscriberAccountUsageDescription	Privacy - TV Provider Usage Description
NSVoIPUsageDescription	Privacy - VoIP Usage Description

A.5 Most common advertising networks in iOS apps

Table A.6 shows all advertising networks in modded and App Store apps present in over 1 000 apps.

Table A.6 Most common ad networks and their counts in modded and App Store apps.

AdNetwork ID	AdNetwork Name	Total	Modded	App Store
4fzdc2evr5.skadnetwork	aarki	29404	12247	17157
wzmmz9fp6w.skadnetwork	inmobi	27022	10923	16099
cstr6suwn9.skadnetwork	google_admob	26943	11113	15830
7ug5zh24hu.skadnetwork	liftoff	26442	10678	15764
4dzt52r2t5.skadnetwork	unityads	26396	10789	15607
v72qych5uu.skadnetwork	appier	26193	10518	15675
yclnxrl5pm.skadnetwork	jampp	26160	10571	15589
2u9pt9hc89.skadnetwork	remerge	26066	10524	15542
hs6bdukanm.skadnetwork	criteo	25978	10523	15455
8s468mfl3y.skadnetwork	rtbhouse	25967	10519	15448
9t245vhmpl.skadnetwork	moloco	25962	10503	15459
c6k4g5qg8m.skadnetwork	beeswax	25792	10410	15382
4468km3ulz.skadnetwork	kayzen_realtime	25752	10486	15266
ppxm28t8ap.skadnetwork	smadex	25671	10386	15285
t38b2kh725.skadnetwork	lifestreet	25553	10393	15160
f38h382jlk.skadnetwork	chartboost	25289	10256	15033
ydx93a7ass.skadnetwork	adikteev	25236	10233	15003
kbd757ywx3.skadnetwork	mintegral	25098	10192	14906
prcb7njmu6.skadnetwork	twitter	24982	10188	14794
3sh42y64q3.skadnetwork	sitescout_centro	24899	10079	14820
5a6flpkh64.skadnetwork	affle_revx	24848	10056	14792
9rd848q2bz.skadnetwork	criteo	24573	10046	14527
424m5254lk.skadnetwork	snapchat	24025	9820	14205
av6w8kgt66.skadnetwork	scalemonk	23940	9831	14109
4pfyvq9l8r.skadnetwork	adcolony	23892	9565	14327
578prtvx9j.skadnetwork	bulbit	23802	9571	14231
mlmmfzh3r3.skadnetwork	triapodi_appreciate	23700	9532	14168
3rd42ekr43.skadnetwork	youappi	23570	9398	14172
22mmun2rn5.skadnetwork	bytedance	23431	9377	14054
klf5c3l5u5.skadnetwork	siftco	23182	9366	13816
s39g8k73mm.skadnetwork	bidease	23175	9354	13821
3qy4746246.skadnetwork	bigabid	23149	9376	13773
uw77j35x4d.skadnetwork	tradedesk	22841	9177	13664

5lm9lj6jb7.skadnetwork	loopme	22033	8911	13122
zq492l623r.skadnetwork	yandex	21990	8810	13180
tl55sbb4fm.skadnetwork	verve_pubnative	21988	8880	13108
m8dbw4sv7c.skadnetwork	dataseat	21829	8833	12996
p78axxw29g.skadnetwork	amazon	21803	8769	13034
v9wttpbfk9.skadnetwork	facebook	21363	8682	12681
44jx6755aq.skadnetwork	personaly	21353	8658	12695
wg4vff78zm.skadnetwork	bidmachine	21251	8584	12667
n38lu8286q.skadnetwork	facebook	21146	8579	12567
glqzh8vgby.skadnetwork	sabiomobile	21132	8542	12590
e5fvkxwrpn.skadnetwork	verizon_yahoogemini	21128	8429	12699
2fnua5tdw4.skadnetwork	adform	21023	8394	12629
n6fk4nfna4.skadnetwork	mediamath	20897	8362	12535
zmvfpc5aq8.skadnetwork	maidenmarketing	20717	8402	12315
3qcr597p9d.skadnetwork	zucks	20596	8269	12327
f73kdq92p3.skadnetwork	spotad	20260	8216	12044
gta9lk7p23.skadnetwork	vungle	19958	7983	11975
238da6jt44.skadnetwork	bytedance	19931	8136	11795
5tjdwbrq8w.skadnetwork	webeyemobile	19670	7857	11813
w9q455wk68.skadnetwork	hybrid_adtech	19455	7805	11650
ludvb6z3bs.skadnetwork	applovin	19362	7804	11558
488r3q3dtq.skadnetwork	adtiming	19299	7861	11438
lr83yxwka7.skadnetwork	imalimedia_apptimus	19209	7807	11402
32z4fx6l9h.skadnetwork	yeahmobi_clicktech	19208	7659	11549
su67r6k2v3.skadnetwork	ironsource	19098	8001	11097
k674qkevps.skadnetwork	pubmatic	18861	7485	11376
v79kvwwj4g.skadnetwork	kidoz	18832	7663	11169
x44k69ngh6.skadnetwork	clearpier	18667	7481	11186
7rz58n8ntl.skadnetwork	mediasmart	18290	7372	10918
9nlqeag3gk.skadnetwork	edge226	18180	7186	10994
f7s53z58qe.skadnetwork	tencent	18153	7279	10874
5l3tpt7t6e.skadnetwork	startapp	18096	7148	10948
mtkv5xtk9e.skadnetwork	maxpoint_interactive	17954	7256	10698
mp6xlyr22a.skadnetwork	clearpier	17901	7179	10722
cg4yq2srnc.skadnetwork	adtheorent	17817	7152	10665
n9x2a789qt.skadnetwork	mytarget	17679	7118	10561
ejvt5qm6ak.skadnetwork	geofy_marsmedia	17524	7043	10481
cj5566h2ga.skadnetwork	engagebdr	17523	7001	10522

pwa73g5rt2.skadnetwork	taptica_tremor	17501	6903	10598
g28c52eehv.skadnetwork	bidslash_zynga	17458	6997	10461
294l99pt4k.skadnetwork	bidswitch_iponweb	17288	6751	10537
24t9a8vw3c.skadnetwork	cheetah_medialink	16855	6734	10121
523jb4fst2.skadnetwork	bidsopt	16545	6623	9922
v4nxqhlyqp.skadnetwork	imobile	16526	6618	9908
ggvn48r87g.skadnetwork	blismedia	16520	6628	9892
xy9t38ct57.skadnetwork	smn_logicad	16226	6507	9719
8c4e2ghe7u.skadnetwork	yahoojapan	15809	6310	9499
ecpz2srf59.skadnetwork	tapjoy	15689	6137	9552
6xzpu9s2p8.skadnetwork	applift	15660	6294	9366
mls7yz5dvl.skadnetwork	bucksense	15568	6238	9330
54nzkqm89y.skadnetwork	tradingworks	15390	6169	9221
feyaarzu9v.skadnetwork	mezzomedia	15315	6137	9178
44n7hlldy6.skadnetwork	spykemedia	15304	6066	9238
9b89h5y424.skadnetwork	frontporch	15294	6131	9163
kbmxgpxpgc.skadnetwork	stackadapt	15255	5984	9271
6g9af3uyq4.skadnetwork	adcolony	15208	6052	9156
275upjj5gd.skadnetwork	mobupps	15183	6106	9077
a2p9lx4jpn.skadnetwork	operasoftware	15044	5539	9505
u679fj5vs4.skadnetwork	UNKNOWN	15039	5938	9101
n66cz3y3bx.skadnetwork	twitter	14990	6008	8982
4w7y6s5ca2.skadnetwork	UNKNOWN	14986	5748	9238
9yg77x724h.skadnetwork	twitter	14820	5931	8889
rx5hdcabgc.skadnetwork	UNKNOWN	14526	5753	8773
qqp299437r.skadnetwork	lunamedia	14421	5735	8686
79pbpufp6p.skadnetwork	UNKNOWN	14344	5726	8618
97r2b46745.skadnetwork	outbrain	14311	5616	8695
r45fhb6rf7.skadnetwork	widenad	14121	5572	8549
737z793b9f.skadnetwork	UNKNOWN	14010	5559	8451
dzg6xy7pwj.skadnetwork	singular_labs	13721	5425	8296
hdw39hrw9y.skadnetwork	singular_labs	13688	5409	8279
y45688jllp.skadnetwork	singular_labs	13668	5397	8271
52fl2v3hgk.skadnetwork	curatemobile	13453	5365	8088
252b5q8x7y.skadnetwork	brave_people	13374	5216	8158
rvh3l7un93.skadnetwork	ventesavenues	13239	5266	7973
nzq8sh4pbs.skadnetwork	adzmedia	13162	5278	7884
74b6s63p6l.skadnetwork	smartyads	12890	4953	7937

7953jerfzd.skadnetwork	mopub	12870	5171	7699
c3frkrj4fj.skadnetwork	viant_adelphic	12769	4970	7799
g2y4y55b64.skadnetwork	globalwide_media	12676	4885	7791
cp8zw746q7.skadnetwork	arpeely	12448	4473	7975
6964rsfnh4.skadnetwork	thingortwo	12203	4642	7561
a7xqa6mtl2.skadnetwork	glispa	12182	4642	7540
84993kbrcf.skadnetwork	tradedoubler_pocketmedia	12144	4620	7524
pwdxu55a5a.skadnetwork	dotc_avazu	12120	4610	7510
x8uqf25wch.skadnetwork	ainiverse_ai4see	11972	4703	7269
47vhws6wlr.skadnetwork	microad	11509	4234	7275
a8cz6cu7e5.skadnetwork	bigo	11374	4226	7148
y5ghdn5j9k.skadnetwork	mediaforce	11306	4137	7169
b9bk5wbcq9.skadnetwork	pipernet	11271	4269	7002
m5mvw97r93.skadnetwork	disciplinedigital	10967	4282	6685
x8jxxk4ff5.skadnetwork	applovin	10856	4351	6505
6p4ks3rnbw.skadnetwork	UNKNOWN	10734	3974	6760
vcra2ehyfk.skadnetwork	adperio_performcb	10634	4121	6513
3l6bd9hu43.skadnetwork	UNKNOWN	10455	4041	6414
7fmhfwg9en.skadnetwork	iqzone	10245	3786	6459
dkc879ngq3.skadnetwork	UNKNOWN	10146	3951	6195
9g2aggbj52.skadnetwork	fyber_digitalturbine	10120	4068	6052
x5l83yy675.skadnetwork	UNKNOWN	9971	3792	6179
s69wq72ugq.skadnetwork	UNKNOWN	9879	3623	6256
krvm3zuq6h.skadnetwork	UNKNOWN	9465	3609	5856
eh6m2bh4zr.skadnetwork	nend	9434	3274	6160
bxvub5ada5.skadnetwork	koneomobile	9322	3442	5880
6v7lgmsu45.skadnetwork	mapendo	9167	3350	5817
vutu7akeur.skadnetwork	line	9034	3097	5937
89z7zv988g.skadnetwork	UNKNOWN	8980	3302	5678
8m87ys6875.skadnetwork	UNKNOWN	8538	3111	5427
9vvzujtq5s.skadnetwork	cygobel	8515	2911	5604
qu637u8glc.skadnetwork	sportradar	8514	3269	5245
dbu4b84rxf.skadnetwork	UNKNOWN	8463	2892	5571
8r8llnkz5a.skadnetwork	UNKNOWN	8455	2985	5470
gvmwg8q7h5.skadnetwork	tnk	8451	3434	5017
cs644xg564.skadnetwork	found	8449	2880	5569
24zw6aqk47.skadnetwork	qverse	8435	2896	5539
hb56zgv37p.skadnetwork	mocaglobal	8378	3010	5368

pu4na253f3.skadnetwork	UNKNOWN	8329	3380	4949
z4gj7hsk7h.skadnetwork	UNKNOWN	8324	3379	4945
m297p6643m.skadnetwork	UNKNOWN	8255	2954	5301
yrqqpx2mcb.skadnetwork	UNKNOWN	8122	3293	4829
bvpn9ufa9b.skadnetwork	unityads	7766	3271	4495
4mn522wn87.skadnetwork	UNKNOWN	7178	2486	4692
gta8lk7p23.skadnetwork	UNKNOWN	6849	2346	4503
nu4557a4je.skadnetwork	jungroup	5820	2257	3563
t6d3zquu66.skadnetwork	UNKNOWN	5098	1804	3294
6yxyv74ff7.skadnetwork	topon	4240	1596	2644
mqn7fxpca7.skadnetwork	adspeq	4031	1351	2680
hjevpa356n.skadnetwork	yieldmo	3978	1583	2395
h65wbv5k3f.skadnetwork	UNKNOWN	3965	1299	2666
x2jnk7ly8j.skadnetwork	UNKNOWN	3815	1554	2261
r26jy69rpl.skadnetwork	mytarget	3249	1290	1959
mj797d8u6f.skadnetwork	bytedance	3191	1212	1979
z959bm4gru.skadnetwork	UNKNOWN	2961	987	1974
k6y4y55b64.skadnetwork	ignitemediahk	2942	1291	1651
g6gcrrvk4p.skadnetwork	UNKNOWN	2730	989	1741
z24wtl6j62.skadnetwork	UNKNOWN	2667	1180	1487
5mv394q32t.skadnetwork	zinkads	2082	800	1282
ln5gz23vtd.skadnetwork	UNKNOWN	2034	752	1282
tmhh9296z4.skadnetwork	adkomo	2033	745	1288
x5854y7y24.skadnetwork	targetmedia	1913	732	1181
7fbxrn65az.skadnetwork	maas_affle	1801	688	1113
fz2k2k5tej.skadnetwork	feedmob	1797	698	1099
sczv5946wb.skadnetwork	mobvista_nativex	1684	605	1079
899vrgt9g8.skadnetwork	bytedance	1631	589	1042
y2ed4ez56y.skadnetwork	UNKNOWN	1553	646	907
67369282zy.skadnetwork	buzzad	1515	559	956
au67k4efj4.skadnetwork	lemmonetmobile	1515	559	956
5ghnmfs3dh.skadnetwork	mobrain	1515	559	956
9wsyqb3ku7.skadnetwork	smartnewsads	1515	559	956
v7896pgt74.skadnetwork	mobione	1509	556	953
2tdux39lx8.skadnetwork	reddit	1509	551	958
eqhxz8m8av.skadnetwork	google_admob	1500	579	921
c7g47wypnu.skadnetwork	crossaudience	1487	547	940
23zd986j2c.skadnetwork	UNKNOWN	1485	664	821

w28pnjg2k4.skadnetwork	fuzeclick	1479	543	936
dr774724x4.skadnetwork	exmox	1474	534	940
qlbq5gtkt8.skadnetwork	boombit	1434	500	934
tvvz7th9br.skadnetwork	UNKNOWN	1416	658	758
cad8qz2s3j.skadnetwork	myappfre	1355	483	872
h8vml93bkz.skadnetwork	jumprampgames	1353	480	873
2q884k2j68.skadnetwork	appalgom	1349	478	871
88k8774x49.skadnetwork	creativeclicks	1349	478	871
pd25vrrwzn.skadnetwork	appsfiremng	1348	474	874
vc83br9sjg.skadnetwork	gmthub	1341	474	867
uzqba5354d.skadnetwork	jumprampgames	1341	474	867
627r9wr2y5.skadnetwork	moburst	1330	471	859
d7g9azk84q.skadnetwork	adupps	1330	471	859
79w64w269u.skadnetwork	adsynthetic	1330	471	859
t7ky8fmwkd.skadnetwork	fascode	1322	467	855
dn942472g5.skadnetwork	motive	1273	449	824
dd3a75yxkv.skadnetwork	dreamperf	1273	449	824
633vhxswh4.skadnetwork	adcountymedia	1273	449	824
h5jmj969g5.skadnetwork	evolvemedia	1272	448	824
y755zyxw56.skadnetwork	bondika	1265	445	820
l6nv3x923s.skadnetwork	jinglemobi	1265	445	820
xx9sdjej2w.skadnetwork	king	1151	405	746
g69uk9uh2b.skadnetwork	UNKNOWN	1112	386	726
33r6p7g8nc.skadnetwork	UNKNOWN	1102	381	721
866k9ut3g3.skadnetwork	amoad	1085	379	706
dmv22haz9p.skadnetwork	mediabuying	1035	358	677
fkak3gfpt6.skadnetwork	olimob	1035	358	677
7k3cvf297u.skadnetwork	gurmob	1035	358	677
gfat3222tu.skadnetwork	appsell	1035	358	677
7tnzynbdc7.skadnetwork	jetads	1035	358	677
3cgn6rq224.skadnetwork	rtbdemand	1035	358	677
b55w3d8y8z.skadnetwork	mobco	1035	358	677
axh5283zss.skadnetwork	yelohi	1035	358	677
8w3np9l82g.skadnetwork	mobwind	1035	358	677
6qx585k4p6.skadnetwork	userbase	1035	358	677
8qiegk9qfv.skadnetwork	kaimedia	1035	358	677
2rq3zucswp.skadnetwork	gameberry	1034	357	677
nfqy3847ph.skadnetwork	simeji	1031	356	675

l93v5h6a4m.skadnetwork	valista	1027	354	673
xmn954pzmp.skadnetwork	adstraction	1027	354	673
zh3b7bxvad.skadnetwork	adsflourish	1027	354	673
t3b3f7n3x8.skadnetwork	cronbay	1027	354	673
jb7bn6koa5.skadnetwork	odenads	1027	354	673
nrt9jy4kw9.skadnetwork	mooko	1027	354	673
z5b3gh5ugf.skadnetwork	dlab	1027	354	673

A.6 Most common signing certificates in iOS modded apps

The most common certificates and in how many apps they are found in total and in each market are included in Table A.7, as discussed in Section 4.3.3. It is worth noting that due to the duplicate apps in some markets, the total number of unique apps signed with each certificate is sometimes smaller than the number of apps signed with each certificate in each market combined.

Table A.7 Most common iOS signing certificates and the markets they appear in.

Company/Developer	Apps	Platin-mods	PDA-LIFE	iOS-Gods	App-Cake	AppDB	IPA-OMTK	Mala-vida	Top-Store	Zeus
Sunshine Insurance Group Co., Ltd.	9 704	0	0	7 027	1	0	229	0	2 468	64
Sun Life Everbright Life Insurance Co.,Ltd	3 667	0	0	3 012	0	0	3	0	651	1
Bank of Chongqing Co.,Ltd.	2 889	0	0	2 828	0	1	42	0	0	28
Zhuhaishi ShijiXintong Network Technology Co., Ltd.	2 390	0	0	2 368	0	0	6	0	0	17
Shanghai Construction No.2 (Group) Co.,Ltd	1 889	0	0	1 884	0	1	4	0	0	1
mu dan	1 575	21	655	2	14	828	51	0	0	44
Rizhao Sunday Power Generation Co., Ltd.	1 463	0	0	1 459	0	0	4	0	0	0
Gac Trumpchi Car Sales Co., Ltd.	1 277	0	0	8	0	0	3	0	1 265	1
Ravand Mohammed Sadeeq	941	0	3	0	0	65	545	0	0	394
China CITIC Bank Corporation Limited	666	0	0	0	0	0	0	0	666	0
Shakeela Shafique Bilal Ahmed	645	0	0	0	0	0	645	0	0	0
Sandhiya Walia	627	0	0	0	0	0	627	0	0	0
convosco UG (haftungsbeschrnkt)	463	0	0	0	0	461	0	0	0	2
Social Governance Coordination Center of Haicang District,Xiamen	267	0	0	0	0	1	0	0	266	0
Beijing Financial Assets Co., Ltd	258	0	0	0	0	2	255	0	1	0
Hussan Mehra	236	0	0	0	0	0	148	0	0	88
Furkan Kale	172	0	0	0	0	0	1	0	0	171
China Continent Property & Casualty Insurance Company Ltd.	139	0	0	0	0	0	138	0	0	1
HUNAN ZHONGQINGLONGTU NETWORK CO.LTD	132	0	0	0	0	0	0	0	132	0

turki alamer	107	0	0	0	0	0	107	0	0	0
Shenzhen Maihong Electronic Co., Ltd.	85	0	0	0	0	0	0	0	0	85
Berke Erdem	63	0	0	0	0	1	62	0	0	0
Soung Saifong	59	0	0	0	0	2	57	0	0	0
John Smith	54	0	0	0	0	16	38	0	0	6
Yvette Testut	49	0	0	0	3	6	1	0	0	43
Luu Anh	44	0	0	0	0	0	44	0	0	0
Resat Yabalak	38	0	0	0	0	0	38	0	0	0
guang zhong fang	37	0	0	0	0	0	0	0	37	0
angel serrano	36	0	0	0	0	35	0	0	0	1
China Mobile Group Jiangsu Company Limited	35	0	0	0	0	7	26	0	2	0
Nazim Ali	32	0	0	0	0	0	32	0	0	0
Elizangela Azevedo	31	0	0	0	0	1	29	0	0	1
Mehsen Mohammad	31	0	0	0	0	9	23	0	0	4
BOE TECHNOLOGY GROUP CO., LTD.	29	0	1	0	0	8	20	0	0	1
feha alsubei	28	0	0	0	0	2	26	0	0	3
Anirba Bose	27	0	0	0	0	0	27	0	0	0
Yusuf Serin	26	0	0	0	0	0	21	0	0	5
Azra Kaya	25	0	0	0	0	0	25	0	0	0
mohamed selim	24	0	0	0	0	0	24	0	0	0
Chrsitian Garcia Delgado	24	0	0	0	0	0	24	0	0	0
Ridvan Topal	24	0	2	0	0	23	1	0	0	1
Fatma Ince	23	0	0	0	0	0	21	0	0	2
Kadam Vohra	22	0	0	0	0	0	22	0	0	0
Bernier R Somerville	21	0	0	0	0	0	18	0	0	3

Appendix B

Modded market operators survey

B.1 Survey instruments

B.1.1 Android modded market operators questionnaire

I. **PLATFORM RELATED QUESTIONS**

1. How was your customer-facing site set up? For example, did you use existing source code, or write your own code with or without the assistance of others? *Open ended*

2. When did your site commence providing third-party Android apps? *Month and Year*

3. Where do you advertise the services provided by your site? *Open ended*

4. How do visitors land on your platform? *Slider 0-100 for each option*

   (a) Search engines

   (b) Advertisements

   (c) Links on other sites

   (d) Visiting the URL directly

   * *Follow up to Q4:* Is the estimate of your visitors a rough impression or it is confirmed by the analysis of your data or by any analytics platform? *Open ended*

5. How do you obtain apps to include on your platform? *Select one only*

   (a) You collect the apps by yourself and release them on your platform

   (b) User submit apps on your platform

   (c) Other (please specify) *Open ended*

   * *Only if Q5(a) is selected:* Briefly explain how you collect the apps? *Select one only*

(a) Going to GitHub and compiling the source code of open-source apps.

(b) Go to Google Play and download them to a handset and pull them out from SD card.

(c) A tool to download these apps from Google Play.

(d) Other (please specify) *Open ended*

* *Only if Q5(b) is selected:* Can people submit apps anonymously to your site? *Yes/No*
* *Only if Q5(b) is selected:* On an average month, how many developers submit apps to your site? *Numeric*

6. On an average month, how many new apps appear on your site each month on your site? *Numeric*

7. On an average month, how many apps are downloaded from your site? *Numeric*

8. Have you been contacted by the developers who uploaded the app to handle bad reviews for their apps on your site? *Yes/No*
   * *Only if Q8 is Yes:* What was the nature of the contact? *Open ended*
   * *Only if Q8 is Yes:* What happened as a result of this? *Open ended*

9. What steps do you perform before releasing an app in your platform? *Open ended*

10. Do you re-sign the APK before releasing the app? *Yes/No*

11. Do you modify the binary/metadata of the APK before releasing the app? *Yes/No*

II. **SECURITY CHECK OF ANDROID APPS**

12. What methods, if any, do you use to test the security of the Android apps on your market? *Open ended*

13. How and why have your security testing methods changed over time, if at all? *Open ended*

14. Have you been contacted by an app developer who claims their app has appeared on your market without their consent? *Yes/No*
   * *Only if Q14 is selected:* What was the nature of the contact? *Open ended*
   * *Only if Q14 is selected:* What happened as a result of this? *Open ended*

15. What steps do you perform when someone reports a security flaw in an app? *Open ended*

III. **GOALS AND MOTIVATION**

16. What are your primary motivations for offering third-party Android apps? *Open ended*

17. How did you became involved in providing third-party Android apps? *Open ended*

18. What are your long-term and short-term goals as an operator for these platforms? *Open ended*

IV. OPERATOR RELATED QUESTIONS

19. On an average month, how many hours do you personally spend maintaining your site and the services it provides? *Numeric*

20. Do you run your site on your own, or with other people? *Select one only*

    (a) On my own, with no others

    (b) With one other person

    (c) With 2-3 other people

    (d) With 3-5 other people

    (e) With 5-10 other people

    (f) With 10+ other people

21. Do you provide third-party services on other websites or operate other third-party services, in addition to this site? *Select one only*

    (a) The one site only

    (b) On one or more additional sites (please indicate which ones) *Open ended*

22. What other online services do you offer? *Open ended*

23. How would you rate your technical skills of operating and running a market? *Slider 0-100*

24. Please select the sentence which best describes your main occupation: *Select one only*

    (a) The provision of third-party Android markets is my main occupation

    (b) Studying

    (c) Working at some other place of employment

    (d) Unemployed

    (e) Retired

    (f) Other (please specify) *Open ended*

25. What is your age? *Select one only*

    (a) 18-24

    (b) 25-34

    (c) 35-44

    (d) 45-54

    (e) 55-64

(f) 65-74

(g) 75-84

(h) 85+

26. Do you identify as? *Select one only*

    (a) Man

    (b) Woman

    (c) Other (please specify) *Open ended*

27. Which continent do you live on? *Select one only*

    (a) Asia

    (b) Africa

    (c) North America

    (d) South America

    (e) Europe

    (f) Australia

28. How do you monetize the market? *Open ended*

29. During an average month, what percentage of your income is provided by your third-party Android site? *Select one only*

    (a) 0-10%

    (b) 11-20%

    (c) 21-30%

    (d) 31-40%

    (e) 41-50%

    (f) 51-60%

    (g) 61-70%

    (h) 71-80%

    (i) 81-90%

    (j) 91-100%

30. During an average month, what percentage of the website's income is reinvested for maintaining and improving your third-party Android site? *Select one only*

    (a) 0-10%

(b) 11-20%

(c) 21-30%

(d) 31-40%

(e) 41-50%

(f) 51-60%

(g) 61-70%

(h) 71-80%

(i) 81-90%

(j) 91-100%

V. **ADDITIONAL QUESTIONS**

31. Do you have any concerns about legal compliance when operating your third-party Android market? *Yes/No*
    * *Only if Q31 is Yes:* How have you come to these views? *Open ended*

32. Is there anything further that you wish to add that you think is relevant to this research? *Open ended*

B.1.2 iOS modded market operators questionnaire

We are researching 'third-party iOS marketplaces'. It would be appreciated if you could answer the following questions.

I. **PLATFORM RELATED QUESTIONS**

1. How was your website set up? For example, did you use existing source code, or write your own code with or without the assistance of others? *Open ended*

2. When did your site commence providing iOS apps? *Month and Year*

3. Where do you advertise the services provided by your site? *Open ended*

4. How do visitors land on your platform? (Slider 0-100 for each option)

    (a) Search engines

    (b) Advertisements

    (c) Links on other sites

    (d) Visiting the URL directly

    (e) From its associated Telegram, Discord, or similar groups, channels, or servers.

* *Follow up to Q4* Is the estimate of your visitors a rough impression or it is confirmed by the analysis of your data or by any analytics platform? *Open ended*

5. How do you obtain apps to include on your platform?

   (a) You collect the apps by yourself and release them on your platform

   (b) Users and/or developers submit apps on your platform

   (c) Other (please specify) _____

   * *Only if Q5(a) is selected* Briefly explain how you collect the apps?

   (a) Going to GitHub and compiling the source code of open-source apps.

   (b) Extracting them from a jailbroken iPhone.

   (c) A tool to download these apps from the App Store.

   (d) Other (please specify) _____

   * *Only if Q5(b) is selected* Can people submit apps anonymously to your site? *Open ended*

   * *Only if Q5(b) is selected* On an average month, how many developers/ users submit apps to your site?

   Numeric _____

6. On an average month, how many apps are updated on your site? *Numeric* _____

7. On an average month, how many new apps are added to your site (not counting updated versions of previously offered apps)? *Numeric* _____

8. On an average month, how many apps are downloaded from your site? *Numeric* _____

9. On an average month, how many visits does your site receive? *Numeric* _____

10. How do you handle user reviews in the apps you host? Do you handle negative reviews in a particular way? *Open ended*

11. What steps do you perform before releasing an app in your platform? *Open ended*

12. Do you modify the apps (IPAs) before releasing them in your marketplace? *Yes/No YES*

    - Please describe in which ways you modify the apps (IPAs). *Open ended*

II. **SECURITY CHECK OF iOS APPS**

13. What methods, if any, do you use to test the security of the iOS apps on your marketplace? *Open ended*

14. How and why have your security testing methods changed over time, if at all? *Open ended*

15. Have you been contacted by an app developer who claims their app has appeared on your marketplace without their consent? *Yes/No*
    YES

    - What was the nature of the contact? *Open ended*
    - What happened as a result of this? *Open ended*

16. What steps do you perform when someone reports a security flaw in an app? *Open ended*

III. **GOALS AND MOTIVATION**

17. Does your marketplace include modified applications? *Yes/No YES*

    - Are they its main focus? *Yes/No*
    - What are your primary motivations for offering modded (modified) iOS apps? *Open ended*

18. How did you became involved in providing third-party iOS apps? *Open ended*

19. What are your long-term and short-term goals for your marketplace? *Open ended*

IV. **OPERATOR RELATED QUESTIONS**

20. On an average month, how many hours do you personally spend maintaining your site and the services it provides? *Numeric* _____

21. Do you run your site on your own, or with other people? *Select one only*

    (a) On my own, with no others
    (b) With one other person
    (c) With 2-3 other people
    (d) With 3-5 other people
    (e) With 5-10 other people
    (f) With 10+ other people

22. Do you provide third-party services on other websites or operate other third-party services, in addition to this site? *Select one only*

    (a) The one site only
    (b) On one or more additional sites *(please indicate which ones)*

    _____

23. What other online services do you offer? *Open ended*

24. Please select the sentence which best describes your main occupation: *Select one only*

    (a) The provision of third-party iOS marketplaces is my main occupation

    (b) Studying

    (c) Working at some other place of employment

    (d) Unemployed

    (e) Retired

    (f) Other (please specify) _____

25. What is your age? Select one only

    (a) 18-24

    (b) 25-34

    (c) 35-44

    (d) 45-54

    (e) 55-64

    (f) 65-74

    (g) 75-84

    (h) 85+

26. Do you identify as? Select one only

    (a) Man

    (b) Woman

    (c) Other (please specify) _____

27. Which continent do you live on? Select one only

    (a) Asia

    (b) Africa

    (c) North America

    (d) South America

    (e) Europe

    (f) Australia

28. How do you monetize the marketplace? Open ended

29. During an average month, what percentage of your income is provided by your third-party iOS site? Select one only

(a) 0-10%

(b) 11-20%

(c) 21-30%

(d) 31-40%

(e) 41-50%

(f) 51-60%

(g) 61-70%

(h) 71-80%

(i) 81-90%

(j) 91-100%

30. During an average month, what percentage of the website's income is reinvested for maintaining and improving your third-party iOS site? Select one only

(a) 0-10%

(b) 11-20%

(c) 21-30%

(d) 31-40%

(e) 41-50%

(f) 51-60%

(g) 61-70%

(h) 71-80%

(i) 81-90%

(j) 91-100%

V. **ADDITIONAL QUESTIONS**

31. Do you have any concerns about legal compliance when operating your third-party iOS market? *Yes/No*

* *Follow up to Q31:* How have you come to these views? *Open ended*

32. Is there anything further that you wish to add that you think is relevant to this research? *Open ended*

Appendix C

Original developers survey

C.1 Survey instrument

C.1.1 Original app developers questionnaire

In our study, we define modded apps as those that have had their code or metadata modified by an unauthorised developer or third-party. This can include changes to advertising libraries, certificates used to sign the apps, app permissions, or even methods in the code to provide users with premium features, or in-app/in-game resources, IAPs, etc.

It would be appreciated if you could answer the following questions:

1. Do you or your organization develop and/or publish mobile applications? *Select one or more*

   (a) Android

   (b) iOS

   (c) Other (please specify) *Open ended*

I. **AWARENESS RELATED QUESTIONS**

2. Before this email, were you aware of the existence of:

   (a) Modded Android apps? *Yes/No*

   (b) Modded Android app markets? *Yes/No*

   (c) Modded iOS apps? *Yes/No*

   (d) Modded iOS app markets? *Yes/No*

3. Before this email, had you or anybody in your organization noticed your apps in app markets? (Yes/No)

   * *Follow up to Q3 if participant selected Yes:*

   Please select the range of your apps you noticed in the app markets. *(Range: 0–100+)*

(a) Modded Android market?

(b) Modded iOS market?

* *Follow up to Q3 if participant selected Yes:* Just to confirm, were these apps there without your, or your organization's, authorization? *Yes/No*
* *Follow up to Q3 if participant selected Yes:* Did you analyze these apps? Did you find any changes? *Open ended*

II. **TOOLS AVAILABLE TO REPORT, PREVENT AND DETECT MODDING**

4. Have you or your organization tried using the modded markets' takedown request forms (e.g. DMCA forms)? If yes, can you please tell us more, such as if the request was acted on, if the app re-appeared, and how long the process took? Please also specify who did the takedown request (e.g. a solicitor, security company, developer). *Open ended*

5. Have you used any tools or coding techniques to try to prevent modding and piracy, if so can you please tell us more? *Open ended*

6. Have you used any tools or coding techniques to try to detect modding and piracy, if so can you please tell us more? *Open ended*

7. Are there any tools or coding techniques you think Google or Apple could provide developers with to help prevent and/or detect modding and piracy? *Open ended*

III. **REVENUE LOSS ESTIMATION AND OTHER IMPACTS OF MODDING**

8. Do you have an estimation of revenue losses to you/your organization caused by modding and/or piracy from: *(Select one or more) (Range: 0-1000+ - Please indicate in USD ($)).*

   (a) Advertising revenue

   (b) In-app purchases

   (c) App sales (e.g. paid apps for free)

   (d) Other *Open ended*

9. Are these estimations based on evidence or data, or a rough guess? *Open ended*

10. Have you also perceived other impacts of modding? For example increased server calls, impact on other users, impact on creators, etc. *Open ended*

IV. **MEASURES TO ADDRESS MODDED APPS**

11. Are you interested in taking any measures to address the availability of modded and pirated versions of your apps on the app marketplaces? *(Select one or more)*

    (a) Yes, I/we **will** request takedown.

(b) No, I **will not** request takedown.

(c) Other (e.g. you have previously involved other service providers (e.g. brand protection or lawyers). *Open ended*

12. In case you want to elaborate any further. *Open ended*

V. **RESPONDENT INFO AND FURTHER COMMENTS**

13. What is your position? *Select one or more*

(a) CEO or similar

(b) Lead developer/ Tech Lead or similar

(c) Senior developer or similar

(d) Medium-level developer or similar

(e) Junior developer or similar

(f) Other (please specify) *Open ended*

14. Roughly how many people work in your company? *(Range: 0-100+)*

15. Is there anything further that you wish to add that you think might be relevant to this research? *Open ended*

16. Would you like to be contacted in the future for: *(Select one or both)*

(a) Notification when our research is published?

(b) Follow up study on takedown requests.

* *Follow up to Q16 if participant selected Yes to any of the options:*
Do you wish to be contacted at the same email address? *(Select one)*

(a) Yes, we can contact you or your organization in the same email address.

(b) We can contact you or your organization in the below email address. (Open ended)

C.2 Original app developers survey codebook

Below is the codebook resulting from the coding of the 717 original app developers' survey responses.

■ Revenue Loss

❇ Ad_revenue_loss

❇ Cloned_apps

❇ Estimation_based_on_data

- Estimation_based_on_rough_guess
- High_percentage_pirate_users
- Little_revenue_loss
- No_estimation
- Paid_app_purchase
- Subscription_or_IAP_revenue_loss

- Modifications
  - Added_malware
  - Changed_advertising
    - added_advertising
    - changed_ad_id
    - removed_advertising
  - Changed_contact_info
  - Changed_permissions
    - added_permissions
  - Feature_mod
    - cheats
    - premium_version_or_features
  - N/A_opensource_app
  - No_analysis
  - Pirated_paid_apps
  - Removed_protections
    - removed_license_checks
  - Unmodified

- Other Impacts of Modding
  - Demoralised
  - Impact_on_other_users
    - unbalanced_online_matches
  - More_errors_crashes
  - None_perceived
    - single_player_offline

- ❖ small_userbase
- ❇ Other_monetisation_methods_unpopular
- ❇ Positive_impacts
  - ❖ more_exposure
- ❇ Reputational_damage
- ❇ Support_requests_modded_apps_or_features_errors
- ❇ Third_party_services_inapp_cost
- ❇ Worried_about_security_of_backend_logic_or_ml

■ Protection Techniques

- ❇ Detection Techniques
  - ❖ installer_package_check
  - ❖ No Detection Techniques
    - ✦ third_party_tools_expensive
  - ❖ proprietary_method
  - ❖ RASP
  - ❖ server_side_checks
  - ❖ third_party_tool
- ❇ Prevention Techniques
  - ❖ google_play_app_integrity
  - ❖ google_play_licensing
  - ❖ No Prevention Techniques
  - ❖ Obfuscation
    - ✦ dexprotector
    - ✦ proguard
  - ❖ prevention_techniques_caused_problems_for_users
  - ❖ proprietary_method
  - ❖ RASP
  - ❖ server_side_checks
  - ❖ third_party_tool
  - ❖ third_party_tools_expensive

■ DMCA-related Actions

- ❇ Future Actions

- ❖ adding_antimod_protections
- ❖ adding_checks
- ❖ dmca_requests

❇ Previous Actions

- ❖ no_dmca_requests
  - ✦ unaware_before_of_modded_markets
  - ✦ unaware_of_dmca_requests
- ❖ removal_google_search_search_results
- ❖ removal_requested
  - ✦ limited_success_dmca_requests
    - ✣ reappeared_after_request
  - ✦ no_success_dmca_requests
    - ✣ cease_and_desist_after_no_success_dmca
    - ✣ dmca_lots_of_proof_needed
    - ✣ dmca_no_response
    - ✣ dmca_time_consuming
  - ✦ removal_requested_through_thirdparty_service
- ❖ success_dmca_requests

■ Google-Apple Solution

❇ Google_Apple_dmca_solution

❇ Search_engines_should_protect

- ❖ search_engines_trademark_protections

❇ Should_not_provide_tools_or_techniques

- ❖ well-known_techniques_would_get_cracked

❇ Should_provide_tools_or_techniques

- ❖ not_allowing_sideloading

❇ Tools

- ❖ Google
  - ✦ google_license_verification_library
    - ✣ lvl_does_not_work
  - ✦ google_play_integrity_api
    - ✣ gpi_api_does_not_work

❇ Unaware_of_current_solutions_provided

- Beliefs
  - Apple
    - apple_does_better
      - have_not_encountered_our_apps_modded_in_ios
  - Good_for_business_and_reach
  - Lack_of_technical_knowledge_to_combat
  - Nothing_can_be_done_not_worth_hassle
    - accepted_modding_part_of_android
    - contact_useless
    - difficult_as_small_developer
    - difficult_to_keep_up
    - more_expensive_dmca_than_accepting
  - Piracy_will_always_exist
    - modded_users_deserve_risks
    - no_big_deal
      - no_big_deal_people_will_buy_if_they_like
      - no_big_deal_people_would_not_pay_anyway
  - Security_risks
  - Threatening_dmca_form
  - Trust_existing_measures_enough
  - Would_dmca_if_automated_or_easy