Cassidy Gibson*, Vanessa Frost, Katie Platt, Washington Garcia, Luis Vargas, Sara Rampazzi, Vincent Bindschaedler, Patrick Traynor, and Kevin Butler

Analyzing the Monetization Ecosystem of Stalkerware

Abstract: Stalkerware is a form of malware that allows for the abusive monitoring of intimate partners. Primarily deployed on information-rich mobile platforms, these malicious applications allow for collecting information about a victim's actions and behaviors, including location data, call audio, text messages, photos, and other personal details. While stalkerware has received increased attention from the security community, the ways in which stalkerware authors monetize their efforts have not been explored in depth. This paper represents the first large-scale technical analysis of monetization within the stalkerware ecosystem. We analyze the code base of 6,432 applications collected by the Coalition Against Stalkerware to determine their monetization strategies.

We find that while far fewer stalkerware apps use ad libraries than normal apps, 99% of those that do use Google AdMob. We also find that payment services range from traditional in-app billing to cryptocurrency. Finally, we demonstrate that Google's recent change to their Terms of Service (ToS) did not eliminate these applications, but instead caused a shift to other payment processors, while the apps can still be found on the Play Store; we verify through emulation that these apps often operate in blatant contravention of the ToS. Through this analysis, we find that the heterogeneity of markets and payment processors means that while point solutions can have impact on monetization, a multi-pronged solution involving multiple stakeholders is necessary to mitigate the financial incentive for developing stalkerware.

Keywords: stalkerware, monetization, mobile security, application analysis

DOI 10.56553/popets-2022-0101

Received 2022-02-28; revised 2022-06-15; accepted 2022-06-16.

1 Introduction

Violence in domestic relationships, also known as intimate partner violence (IPV), involves physical, sexual, or psychological harm done to a partner in an intimate relationship. While both women and men are potential victims of IPV, women are disproportionately affected, with the World Health Organization estimating that 27% of women worldwide aged 15-49 who have been in a relationship have been subject to a form of physical or sexual violence by their partner [43], an issue that has only been exacerbated by the COVID-19 pandemic [19].

Unfortunately, technology has enabled IPV in numerous ways, perhaps most notably by facilitating pervasive surveillance of intimate partners. Smartphone apps called *stalkerware* allow for the collection of personal details such as web searches, location, messages, photos, and other information, making this information accessible to an abuser while hiding their functionality from the victimized partner. Past research has considered the unique threat landscape faced by survivors of technology-enabled abuse [16, 39], characteristics of stalkerware and related apps [11, 35], clinical approaches to aid survivors [15, 20, 41] and understanding the motivations of abusers [9, 40]; however, to date, technical analysis of apps has largely been limited to relatively small corpora [21, 38]. Moreover, there has been little examination of how developers of stalkerware financially benefit from their harmful software.

In this paper, we perform the first large-scale technical analysis of stalkerware to characterize and analyze the monetization mechanisms used by these apps for the developer's financial gain. We retrieve over 6,400

***Corresponding Author: Cassidy Gibson:** University of Florida, E-mail: c.gibson@ufl.edu
Vanessa Frost: University of Florida, E-mail: vfrost@ufl.edu
Katie Platt: University of Florida, E-mail: katieplatt@ufl.edu
Washington Garcia: University of Florida, E-mail: w.garcia@ufl.edu
Luis Vargas: University of Florida, E-mail: lfvargas14@ufl.edu
Sara Rampazzi: University of Florida, E-mail: srampazzi@ufl.edu
Vincent Bindschaedler: University of Florida, E-mail: vbindschaedler@ufl.edu
Patrick Traynor: University of Florida, E-mail: traynor@ufl.edu
Kevin Butler: University of Florida, E-mail: butler@ufl.edu

Android apps identified as stalkerware and collected by the Coalition Against Stalkerware [4] over a 16-month period (July 2020 - November 2021), decompile these apps to perform static analysis over them, and examine their use of monetization strategies. In particular, we focus on their use of ad libraries, in-app payments, and external websites based on code and data recovered from these apps. We also focus on understanding the implications of Google's updated terms of service, effective October 1, 2020, that effectively ban apps deemed to be stalkerware from the Google Play Store [17] and in theory, from being able to use in-app billing as a means of collecting revenue [18]. We thus make the following contributions:

- **In-App Advertising Libraries in Stalkerware:** Compared with normal apps, we discover that substantially fewer stalkerware apps make use of ad libraries. Of the apps that do, the vast majority of apps (99%) use Google AdMob, and most apps use this ad library to the exclusion of all others.
- **Payment Processors:** We demonstrate that over time, payment processing mechanisms have become increasingly diverse. While PayPal and AdMob represent monetization services used by the majority of stalkerware apps, credit card processing, external payment processors such as Square and Stripe, and even cryptocurrency are being leveraged.
- **Terms of Service Analysis:** We use crowdsourced data from VirusTotal to approximate the date that a stalkerware sample was first seen in the wild, and correlate this data with app monetization behavior before and after Google's changes to the Play Store Terms of Service (ToS). We find a measurable and significant change in monetization strategies, with Google Play's in-app billing libraries present in 57% of apps dated prior to the ToS change, while only 15% of stalkerware apps first seen after the October 2020 Play Store changes contain this code. However, we also find over 141 apps identified as stalkerware are still actively available on the Play Store and use the same monetization strategies; furthermore, these apps often contravene the Terms of Service in their behavior, as discovered through app emulation.

As such, we surmise that specific steps can be immediately taken to affect the revenue stream of many stalkerware apps, but given the heterogeneity of payment processors and markets, a multi-faceted solution by multiple stakeholders is necessary to significantly affect the monetization ecosystem.

The rest of the paper is structured as follows: Section 2 provides background on stalkerware and monetization strategies; Section 3 describes the analysis techniques we use throughout this study; Sections 4 through 6 describe our methodology and results for analyzing ad libraries, in-app payments, and analysis of apps in the wake of Play Store Terms of Service changes, respectively. Section 7 summarizes our recommendations and describes threats to validity of the study; Section 8 highlights related work, and Section 9 concludes.

2 Background

Domestic abusers are relying more frequently on technology to track and monitor their victims' behaviors. In a 2014 survey conducted by the National Network to End Domestic Violence, over 50% of survivors reported being tracked or monitored by smartphone apps [31].

Often times, these apps may be downloaded onto a user's phone with or without their permission and/or their knowledge. Even users who knowingly download these apps are often doing so under the threat of coercion. Once these tracking applications are on their victims' devices, domestic abusers can access a variety of sensitive information, such as location and communication data [11, 20, 32].

Many stalkerware apps are created for the explicit purpose of tracking someone, and may be subtly marketed towards individuals who, for example, suspect their spouse of infidelity. The legality of these applications is controversial, with privacy advocates pointing out that creating these tools enables abusers to stalk their victims [24]. Some apps were originally created with legitimate uses in mind that were then repurposed by abusers [11], e.g., tracking the location of potentially lost or stolen devices. These are known as *dual-use* apps, as abusers can take advantage of shared accounts (or simply coerce login information from their victim) to enable remote surveillance. In this paper, we consider only apps identified as stalkerware by the Coalition Against Stalkerware. We further discuss our repository in Section 3.

While the original purpose of these apps may vary, their ultimate uses have led app stores to take measures that mitigate the risks associated with these apps. Some app stores have started explicitly banning stalkerware applications in their terms of service [17]. The effects of these bans are discussed further in Section 6. However, as we discuss later in Section 3, many abusers are able to find applications that are still accessible within app stores. Additionally, some stalkerware authors may choose to publish their apps on third-party app stores or sell them directly from the app's website such as shown in Section 6.

Once the app is downloaded onto the victim's phone, the abuser does not need to access the app directly again. Instead data collected by the app is uploaded to the company's server where the abuser is able to access the information at will, through tools provided by the company such as a website or a companion app.

2.1 Monetization Schemes

While some apps may require the user to pay a flat fee prior to installation, the vast majority (over 90% on the Apple App Store and 95% on the Google Play Store) are free to download [22]. App developers thus have three primary means by which they monetize their product and generate revenue:

1. *In-App Purchases*: These purchases are often referred to as micro-transactions and provide additional functionality or resources to the user based on the amount paid. The price of these purchases often start as low as $0.99 USD.
2. *Advertisements*: Revenue generated by ads can be broken into three subgroups: (A) Per-Thousand Impressions, where the app owner is paid by how many times an ad is seen by their users; (B) Cost Per Click, the amount paid to the app owner when a user engages with an ad by clicking on it while using the app; and (C) Cost Per Action, in which the app owners get a portion of what app users spend when they follow the ad to another site. These actions often generate small amounts of revenue that add up over time. However, using mobile ads also allows ad libraries to access any information about the user that the app has permission to access. The data can then be used to create more targeted ads for the user. Since these ads are targeted, users may be more likely to engage with the ad, generating more revenue for the app.
3. *Subscription*: Subscription-based apps are often free to download, but contain features that must be unlocked. In-app subscriptions are conceptually similar to in-app purchases, with the difference being a recurring revenue model and the cessation of functionality if the subscription is not renewed. Alternatively, out-of-app subscriptions require the user to access an external location, often a website, where the product is paid for and activated.

In our study, we focus primarily on the transaction of money occurring within stalkerware apps. We believe that by understanding how stalkerware generates revenue, we can create more cohesive plans to shut down the development and use of these apps, as removing monetization sources will render their continued operation economically infeasible. Since it has a peripheral relationship to in-app purchases, we briefly examine on external websites that may be used to pay for these apps (Section 3). Otherwise, the focus of this paper is generation of money through the app itself. We broadly refer to this as In-App Monetization.

3 Stalkerware App Analysis

3.1 Stalkerware Threat List

The Coalition Against Stalkerware is an international collaboration between a diverse set of partners including IT security companies, domestic violence survivor networks, organizations that work with perpetrators, digital rights advocacy groups, and law enforcement [4]. The Coalition maintains the *Stalkerware Threat List* (STL), a repository of malware samples that partner organizations have identified as stalkerware through their threat intelligence networks.

We created a local mirror of the repository with additional metadata discovered through processing these samples and correlating this information with other sources. Our local mirror covers all samples uploaded to the STL between July 14, 2020 and November 3, 2021, covering a total of 8195 samples. While some of these samples represent Windows executables and other files, the vast majority of samples are targeted towards deployment on Android devices.

For each of the 6439 samples identified as an Android application package (i.e., an APK file), we extracted metadata about the sample and decompiled the binary applications to source code using the jadx decompiler [36]. The metadata included the application manifest file to check permission use, package name, and the Android SDK version identified as a target for the app. In the case that jadx was unable to decompile the APK, we instead use JEB Pro [33] to decompile the application. When examining the output of these two decompilers, we found jadx was able to recover all fea-

keyword	relevant	filtering
access_token	no	n/a
amex	yes	blocklist
billing	yes	allowlist
bitcoin	yes	allowlist
cardano	no	n/a
coinbase	no	n/a
discover	yes	both
dogecoin	no	n/a
ethereum	no	n/a
mastercard	yes	blocklist
monthly payment	yes	n/a
monthly plan	yes	n/a
payment	no	n/a
paypal	yes	blocklist
ripple	no	n/a
satoshi	no	n/a
square	yes	allowlist
stripe	yes	allowlist
subscription	yes	allowlist
transaction	no	n/a
venmo	yes	blocklist
visa	yes	allowlist

Table 1. To identify functionality within the source code of the decompiled stalkerware samples, we used keywords. The column 'relevant' shows whether the keyword was useful to our analysis and the filtering strategy used to remove false positives from the search.

tures needed for our analysis in the apps it was able to decompile. JEB Pro did not recover any additional application features that we targeted in our analysis. As such we only applied JEB Pro to the apps that were obfuscated and could not be completed by jadx.

In total, 733 samples could not be decompiled by jadx and were run through JEB Pro. From those 733 samples, JEB Pro was able to successfully decompile 726 of them. Overall, we were able to successfully recover source code from 6432 samples. For the rest of the paper, when we refer to *samples*, we are referring to this corpus of successfully decompiled Android apps.

3.2 Keyword Analysis

We examine the source code of stalkerware in order to determine how they generate their revenue. Despite the absence of direct prior research on popular in-app payment methods, we took inspiration from Cardpliance [29], Spamalytics [23], and LibRadar [28] to build a comprehensive list of payment methods in mobile apps. The result of this was a list of sensitive keywords (shown in Table 1) to search for in the source code of the decompiled apps, implemented as a recursive grep search to pull the information. These search results were stored in Apache Spark dataframes.

For each instance of a keyword match, a row lists which app version, file location, line number, and line the keyword was found in. For each keyword, false positives were identified by manually reviewing the source code, and filtered from the dataframe. By manually reviewing the source code, we were able to identify unique keywords that were unlikely to generate false positives (such as "PayPal"), which were filtered by blocklisting certain lines and locations. Common keywords that were often present in the case of false positives were filtered through allowlisted lines and locations. This allows our analysis to be as accurate as possible, without compromising the scale of the data a keyword search provides. Information included in the AndroidManifest.xml file of each app was also extracted and stored in a dataframe, then joined with the keyword data.

We checked the Google Play Store for each stalkerware package name and used the "APK Downloader" Google Chrome extension to download 145 apps that were found. These were then uploaded to APKLab.io, an online mobile app threat assessment platform, and analyzed with its tool [8]. In particular, we focus on the dangerous permissions and deprecated encryption algorithms used by the apps. Additional information provided by the app page on the Google Play Store was also recorded, such as the pricing of in-app purchases and developer identity.

Using these methods, we are able to examine trends over time using the required fields that denote which SDK version the apps are built for (such as platformBuildVersionCode and targetSdkVersion) [6], identify some samples as different versions of the same app, and determine several monetization schemes used by the apps. For instance, we were able to determine which apps support (and use) third-party payment processors (such as PayPal and Stripe), which apps operate on a subscription model, and even identify specific payment accounts that are used by the stalkerware apps. The results of this analysis are discussed in Section 5.

3.3 Emulation

Of the 145 stalkerware apps that were still available on the Google Play Store as of August 2, 2021, we designated three researchers to manually review each app page to gather information on the app's high-level behavior (such as advertised capabilities, in-app purchases, and user reviews) and compare it to the Coalition Against Stalkerware's definition of stalkerware.[1] 129 of those stalkerware apps still remain on the Google Play Store as of October 30, 2021. Using the Google Play Store app description and mentions of the app hiding icons and tracking, we identified 16 apps to emulate.

We emulated these 16 apps on the Android Studio Emulator using a Google Pixel 2 virtual phone, as well as viewed the apps by installing them on a physical Google Pixel phone. While some apps can detect when they are being emulated and may subsequently change their behavior, we found no differences in the apps between the two platforms other than minor networking and connectivity errors.

While investigating each app, we attempted to identify the creator of the app, whether a group or individual, the app's monetization scheme, and to understand the full capabilities of the stalkerware. Furthermore, we noted all possible in-app purchases and what capabilities these purchases added to the app. Most of these transactions allowed users to remove ads from the app, pay for a premium version of the app, or prompted the user to upgrade to a paid version at the conclusion of a free trial period. As of October 2020, Google updated their terms of service to require apps presently running on the Android phone to present a background icon in the notification bar. Because of this, we also noted what the icon looked like for the app, if a notification was displayed while the app was running in the background, and if the app icons appeared to be obscured, hidden, or camouflaged in any way.

3.4 Ecosystem Characterization

To characterize the financial ecosystem of stalkerware, we investigate the different monetization schemes of stalkerware apps. As shown in Figure 1, we were able to identify monetization schemes for 67% of samples including the use of in-app advertisements, monetization through in-app purchases, and links to out-of-app monetization mechanisms from external websites. In the following sections, we describe our analysis of these monetization techniques and our results. Note: we limit our inquiry to data found through analysis of the samples; we do not consider purchases through third-party sites, for example, that the sample does not connect to, or payment channels such as offline merchants providing access codes. We discuss these details further in section 7.

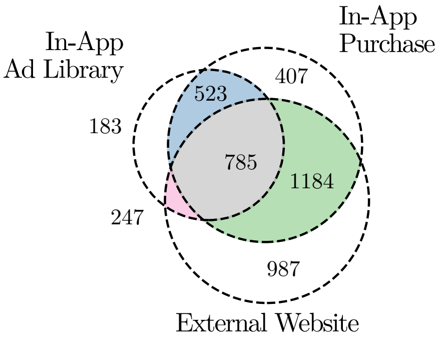


Fig. 1. Number of samples in each monetization type (and intersections) from our analysis. The majority of stalkerware apps in our analysis (4,316, or 67.1%) use a mixture of monetization schemes within their apps. The other 2,116 (or 32.9%) were not identifiable. However, they could be making money outside of their app such as making the abuser purchase the app before download.

3.5 Data Monetization

After analysis on monetization through the use of ad libraries, we also wanted to consider the use of data monetization libraries within our corpus. Through a cursory search, we found evidence of over 20 libraries that monetize data by collecting it to later sell to a third-party. However, these SDKs have a high barrier for entry to use. Most of these sites require to sign up for the service, with a package name to the app the developer wants to add the service to.

From these data monetization libraries, we found that six were dashboards that developers could use to manually upload data to sell to distributors. Considering that stalkerware allows more permissions than normal applications, these applications have a surplus of information gathered for these dashboards. However, analyzing this revenue source is elusive, since this process is done manually, by the stalkerware app developer. In absence of a standard approach in literature on in-app data monetization, more extensive analysis is needed to identify the libraries and tools commonly used for data monetization and how to identify them. Furthermore, the rise of server-to-server data transfers means

[1] The coalition's definition of stalkerware is "software, made available directly to individuals, that enables a remote user to monitor the activities on another user's device without that user's consent and without explicit, persistent notification to that user in a manner that may facilitate intimate partner surveillance, harassment, abuse, stalking, and/or violence" [12].

that we are unable to effectively characterize these relationships from the apps alone [42]. Since location and other personal information is accessible and uploaded by these services to dashboards, stalkerware developers have a trove of information that they could monetize. Offline analysis of how stalkerware monetizes this data is future work.

4 Ad Library Analysis

Stalkerware apps can deploy a diverse set of vectors to enable monetization of the app. One vector is the use of Android ad libraries, which to date are not well understood in the context of stalkerware apps. Ad libraries could be a preferred technique of stalkerware app developers, as the app can be monetized without relying on in-app purchases. Thus we examine the use of Android ad libraries in our corpus of apps and compare their use to normal apps to see if there is a difference. If ad libraries are a potent technique by which these apps are monetized, it offers a viable strategy to plug the monetization pipeline, since it is feasible to moderate the serving of ads [7].

4.1 Methodology

Since our app analysis focuses on Android apps, we leverage a list of 63 Android ad libraries developed from a recent investigation that looked specifically into the most popular ad libraries used by apps in the Google Play Store [7]. The list enables us to compile regular expressions for each ad library, and likewise perform a keyword search over the import statements of recovered source code from the 6432 samples of stalkerware and 442 samples of normal "benign" apps that came from the Google Play Store's Top 500[2] list. Furthermore, to ensure the samples that imported the ad libraries were using them, we confirmed that the respective links for these ad libraries were present URLs collected during the dynamic APKLabs analysis.

In our comparison of the two groups, we take a random sampling of 250 apps that use ad libraries both from the benign apps and stalkerware apps. These samples ensure that our analysis between the two app categories is not biased due to an imbalance in the dataset sizes. Our analysis assumes that if an app developer is importing a relevant ad library, they are using it for in-app monetization through ad revenue or an option to remove ads through an in-app purchase.

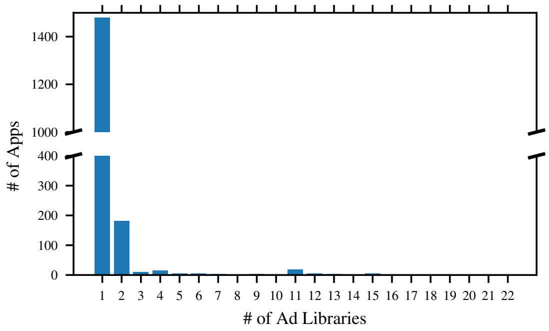


Fig. 2. Histogram of the frequency of ad library counts in stalkerware apps. 85% of Stalkerware apps only leverage a single ad library.

4.2 Results

Our keyword search offers a first look into the distribution of ad library use in stalkerware apps. In total, 1,738 samples (27% of the corpus) leverage 35 of the 63 ad libraries investigated, as evidenced by the decompiled import statements, with 183 apps relying *only* on ad libraries to generate their revenue . We take a deeper look by plotting the distribution of ad library use, shown in Figure 2. The majority of apps (1,479) use only one ad library. The diversity of ad library use diminishes quickly, with only 182 apps importing two ad libraries, and a total of only 77 apps using three or more.

We can pin-point which ad libraries are responsible for the app monetization by comparing the frequency distribution of the top ad libraries seen in our corpus, shown in Figure 3. The majority of apps rely on Google AdMob for their ad monetization, accounting for 1,720 apps, or 99.0% of all samples that implemented ad libraries. In fact, 1,468 apps rely on *only* Google AdMob(shown in figure 3).

This is followed by the second most popular ad library in our corpus, Facebook Audience Network (FAN), which was seen in 185 of apps. When apps leverage ad libraries other than Google AdMob, they are often diverse. For example, of the 37 ad libraries used in apps, 32 (or 86%) were used in conjunction with some other ad library in the same app.

[2] https://www.appannie.com/

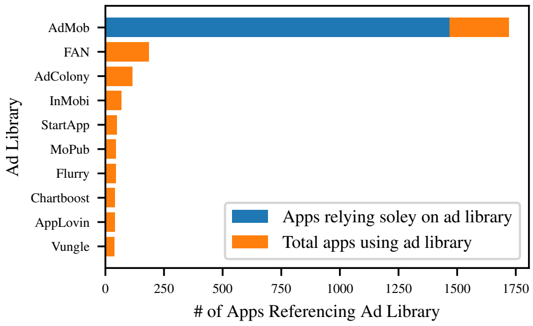


Fig. 3. Distribution of the top ad libraries used in stalkerware apps. We show that most apps rely on *only one* ad library, Google AdMob.

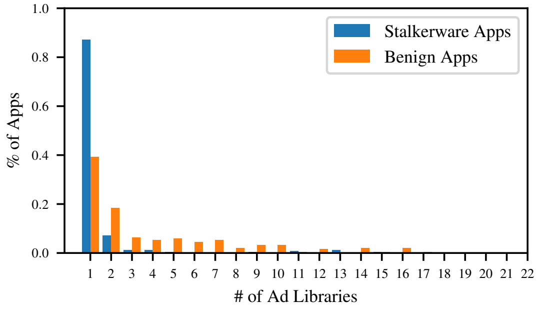


Fig. 4. Number of apps that use one or multiple ad libraries. It is much more common for benign apps to use multiple ad libraries at a time and not rely on a singular ad library to generate their revenue.

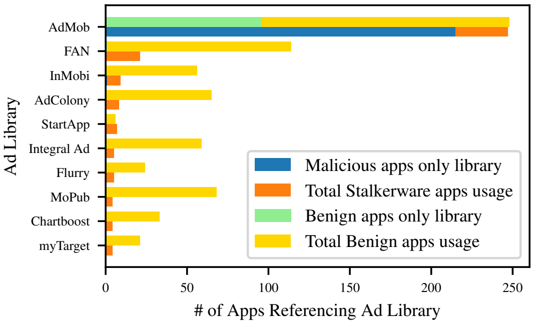


Fig. 5. Distribution of apps that implement a certain ad library. Google AdMob is common in both normal apps and stalkerware, but normal apps often use other ad libraries in addition to Google AdMob.

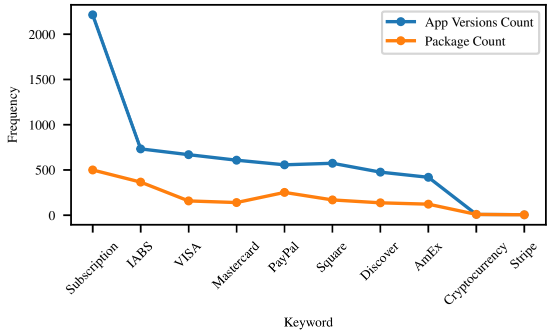


Fig. 6. Frequency of keywords in app versions (blue line) and packages (orange line). Keywords inform the potential behavior of stalkerware apps, as well the various in-app payment options they offer.

We compare these findings in stalkerware with identical tests done on the group of benign apps. Using our keyword search on these popular apps, we identified 263 apps (53% of the normal apps) that use ad libraries and look into the distribution of ad library use in these normal apps. Most importantly in Figure 4, we can see it is much more common for a benign app to use multiple ad libraries and not rely on a single one to generate their revenue. We can see this further reflected in Figure 5. While Google AdMob is prevalent in both groups, Google AdMob is relied upon as a stalkerware app's ad library more than twice as frequently as normal apps (i.e., 215 times by stalkerware, and only 96 times by normal apps). As such, though normal apps use the same ad libraries as stalkerware, stalkerware apps heavily rely on Google AdMob, which often acts as their only ad library to generate revenue in the app.

5 In-App Payments

Due to the increasing number of payment processors on mobile platforms, stalkerware developers have many options for monetization through the app directly. To understand the ways in which in-app payment mechanisms are used for monetization, we leverage the keyword analysis approach described in Section 3. Our in-app keywords are selected to uncover payment options across the corpus of samples, which are referenced in the code itself. This differs from references in URLs and external pages, which we discuss in later sections. Our goal is to investigate support for third-party vendors (such as PayPal), direct payment processing (credit/debit cards), cryptocurrency, and Google's In-App Billing Service (IABS).

5.1 Results

To investigate the latent monetary interaction between abuser and stalkerware app developer, we plot the frequency of keywords across app versions and packages,

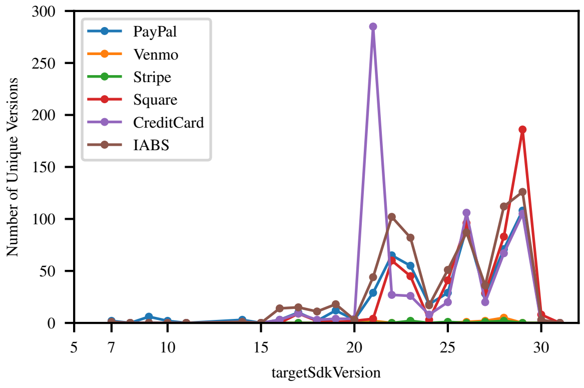


Fig. 7. In-app payment processor support across samples, grouped by the app's targetSdkVersion field.

shown in Figure 6.[3] In total, 2,899 apps make a reference to some in-app billing option, representing 743 packages. Over 2,200 app versions across 500 packages make references to a subscription model. Likewise, some type of billing is referenced in 733 app versions across 366 packages. These general terms offer a high-level view of how abusers interact with the software (e.g., the apps offer a service which necessitates constant support from the abuser). References to different cryptocurrencies exist but are rare, with only nine apps making references to these financial vehicles. The more popular options are Visa, Mastercard, and PayPal, with 669, 608, and 557 app counts, respectively. The use of external payment processors is evident, with Square and Stripe referenced 574 and 5 times respectively. IABS is the most commonly used method of processing payments within these apps, shown by the "billing" keyword, with 733 samples explicitly supporting it.

The results of Figure 6 imply that in-app billing is common among stalkerware apps. In fact, there is evidence of in-app payment to third-party payment processors, accounting for 689 samples that support at least one third-party processor. There is a potential interaction between the targeted Android SDK build version and availability of third-party vendor services. To understand this better, Figure 7 shows a comparison of vendor counts across the extracted Android SDK version which is denoted by the targetSdkVersion in the Android Manifest. This number expresses the SDK version which the app was prepared for (e.g., targetSdkVersion = 29 represents Android 10).[4] Earlier versions demonstrate little monetization apart from IABS. However, after version 20, we see that third-party payment vendors became more prominent, potentially due to the increasing availability of their payment libraries to developers. This supports the idea that as third-party payment options became more diversified on Android, so did the payment options for stalkerware app developers. Likewise, we can say that stalkerware developers are quick to adopt new payment options. These options lean towards Square and IABS in later versions (25+), although credit card options are still prominent.

Looking at Figure 7, we see that stalkerware developers rely on third-party vendors for monetization. This evidence can persuade third-party vendors to block certain developers from the use of the monetization service. We investigate the feasibility of mitigation by manually reviewing the terms of service (ToS) for each third-party vendor and find that most vendors address and oppose stalkerware either implicitly (by not allowing any sort of criminal activity, which includes stalking) or explicitly (such as in Google's Play Store policy, which includes a definition of stalkerware).

6 Effects of Terms of Service Change

To thwart stalkerware, some companies have implemented changes to their policies and terms of service (ToS). In particular, Google updated the Play Store ToS on September 16, 2020 to prohibit stalkerware apps and require that apps monitoring or tracking a user's behavior be more transparent. The updated ToS went into effect on October 1, 2020 [17]. The relevant text from the updated ToS is the following:

> Non-stalkerware apps distributed on the Play Store which monitor or track a user's behavior on a device must minimally comply with these requirements:
> 1. Apps must not present themselves as a spying or secret surveillance solution.
> 2. Apps must not hide or cloak tracking behavior or attempt to mislead users about such functionality.

[3] We discovered in our corpus of stalkerware samples that some samples share a common package name in their Android manifest files and thus represent different versions of the same app. We differentiate between app *versions* and the unique app package names (or in short, *packages*) in this analysis.

[4] See https://developer.android.com/studio/releases/platforms for more information about Android SDK versions and their relationship with Android version numbers.

3. Apps must present users with a persistent notification at all times when the app is running and a unique icon that clearly identifies the app.
4. Apps and app listings on Google Play must not provide any means to activate or access functionality that violate these terms, such as linking to a non-compliant APK hosted outside Google Play.

To date it remains unclear what effect this ToS change has had on the stalkerware app ecosystem, particularly the monetization schemes they employ. To study the effect of the ToS change, we perform an analysis comparing Pre-ToS and Post-ToS apps in our corpus, based on their most common monetization keywords. If monetization strategies have significantly shifted in response to the updated ToS (or over time), we expect to see evidence of it when comparing Pre-ToS apps and Post-ToS apps. This should be most evident in those Post-ToS apps that remain on the Google Play Store today.

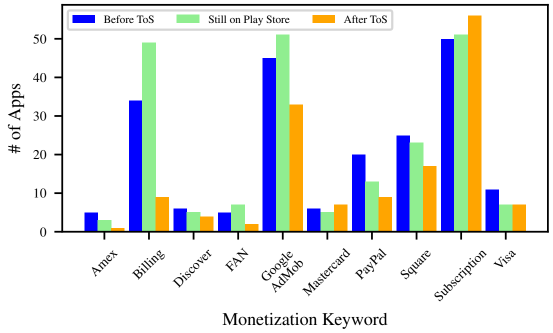


Fig. 8. Longitudinal analysis of common monetization keywords for Pre-ToS apps (group one), apps still on the Google Play Store (as of October 2021) (group two), and apps first seen Post-ToS implementation, which are not on the Google Play Store (group three). Post-ToS apps evidence a shift in monetization behavior compared to apps still on the Google Play Store.

6.1 Methodology

To perform the aforementioned analysis, we classify 1,031[5] apps from our corpus as either Pre-ToS or Post-ToS. We use the app's earliest submission date from VirusTotal to use as a proxy for the first published date. Concretely, we partition the 1,031 apps into three categories:

1. apps known to VirusTotal before the ToS change;
2. apps known to VirusTotal before the ToS change *and* that are still on the Google Play Store as of the time of writing; and
3. apps first known to VirusTotal *after* the ToS change and *not* on the Google Play Store.

The monetization keywords extracted from group (1) informs what the monetization landscape was before the ToS update. Similarly, monetization keywords from group (2) informs how apps are monetized under scrutiny of the updated Google ToS. Billing keywords from group (3) can inform the monetization Post-ToS change and away from the Google Play Store ecosystem. We emphasize however that the VirusTotal first seen date is only an approximation of when an app was initially developed and launched. In other words, this analysis assumes that the earliest submission date from VirusTotal is correlated to the contractual atmosphere in which it was developed and released (e.g., with the updated Google Play Store ToS in mind or not).

To enforce temporal locality in the analysis, we only consider apps that were first seen a year before or after the ToS implementation date (i.e., between October 1st 2019 and October 1st 2021). After this filtering, we are left with 584 apps (representing 145 packages). We are only interested in apps that meet the group conditions, which left 471 apps in total. However, it is possible for older versions of apps, which existed Pre-ToS, to be initially uploaded to VirusTotal *after* the ToS change went live, perhaps due to delayed spread on other ecosystems. These samples introduce a confounding factor since they do not reflect true Post-ToS monetization strategies. We filter out these samples by querying the package associated with each app in group (1), and then retrieving the set of Android SDK target versions associated with their package. The maximum version of each package represents the latest theoretical version a package reached before the ToS was changed. In group (3) (Post-ToS group), we remove apps with a version less than this maximum. Using this strategy, we removed eight of the apps in the analysis corpus. In total, we found 112 apps that existed prior the ToS change (group (1)), 59 apps that existed prior the ToS change and are still on the Google Play Store (group (2)), and 300 apps that were *only* seen after the ToS change, outside the Play Store (group (3)).

[5] These are samples for which we were able to retrieve a date from their VirusTotal analyses.

6.2 Results

Among these apps, we are interested in their monetization strategies, which we posit is indicated by the discovery of ad libraries, monetization services, or generic billing keywords in the decompiled app code. In Figure 8, we show the frequency of the most common keywords among apps in each group. To enable comparison, we uniformly randomly sampled 59 apps from each group (the lowest count for group (2)). The figure shows a shift in monetization strategy caused by the change in the TOS, evidenced by the terms Billing, Amex, Discover, Facebook Audience Network, Google AdMob, Paypal, Square, and Visa. There is a decrease in frequency for both Play Store and Post-ToS groups on certain keywords, such as AmEx, Discover, PayPal, Square, and Visa, which are always lower than the Pre-ToS baseline. This is in contrast to keywords like Mastercard and Subscription, which are more prominent in apps Post-ToS. Given the granularity of our analysis, it is not possible to conclusively determine the precise reason for this shift. However, there is a clear interaction between the presence of keyword terms and the ToS change, which suggest a shift in monetization strategies occurring around the time that the Google Play Store's updated ToS took effect. This observation can inform and be the starting point for future monetization studies that investigate runtime behavior of these apps.

Furthermore, since the Billing keyword was prominent in the analysis of Pre-ToS and Play Store apps, but not Post-ToS apps, we posit that this term is related to the use of Google's in-app billing service, which can only be accessed if the app is hosted on the Google Play Store.[6] We verified this by querying the decompiled code of apps in each group for InAppBillingService (IABS), the Java class which interacts with the Google billing service. The IABS was present in 57% of Pre-ToS apps, 83% of Play Store apps, and 15% of Post-ToS apps. This result is in line with the frequency of Billing in Figure 8. Notably, IABS is only useful if the app is hosted on the app store. This may indicate that 15% of the Post-ToS apps were originally meant for the Google Play Store, but were re-packaged and released elsewhere due to the ToS change.

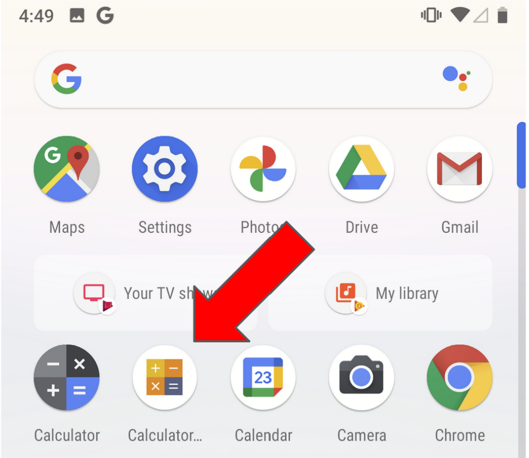


Fig. 9. Example of app camouflaging (indicated by red arrow). The icon of the emulated app Flash Keylogger [5] appears similar to the default Android calculator app. At the time of analysis the app was advertised as a "monitoring app for family members". It has since been removed from the Google Play Store [10, 30].

6.3 Emulation

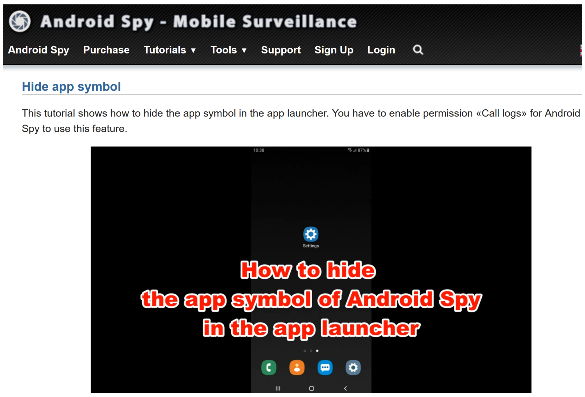
Following the analysis on the Terms of Service change, we emulate 16 stalkerware Play Store apps representative of our repository to confirm our findings and retrieve information regarding in-app purchases, capabilities of the app, areas in which the apps complied with (or directly infringed upon) the Google Play Store Terms of Service, and the monetization schemes used by the apps.

Of the 16 apps we analyzed, 9 of them were completely free, without advertisements or an upgrade option. An additional 3 were free, but offered the ability to pay to remove advertisements or upgrade for additional features. Another 2 apps offered a free trial period, and required either a subscription to continue using the app or a one-time upgrade to a pro version.

14 of these apps had a clearly displayed, unique app icon. Of the two that did not, one we were unable to emulate due to it being a paid app, and one allowed the user to camouflage the app as a calendar, calculator, or note-taking app as shown in Figure 9. When accessed, the user was required to enter a pin number in order to open and use the app.

Similarly, 12 out of the 16 apps did not display an app icon to designate that the app was running in the background as per the Play Store Terms of Service. Only 2 properly and clearly displayed notifications indicating that the app was running and collecting data in the background, while another implemented an icon that

[6] https://stuff.mit.edu/afs/sipb/project/android/docs/training/in-app-billing/preparing-iab-app.html



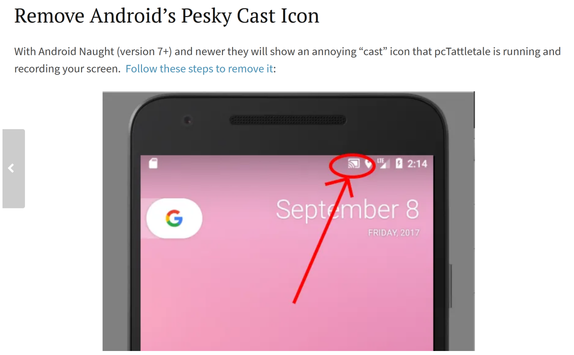
Android Spy - Mobile Surveillance

Android Spy Purchase Tutorials ▾ Tools ▾ Support Sign Up Login

Hide app symbol

This tutorial shows how to hide the app symbol in the app launcher. You have to enable permission «Call logs» for Android Spy to use this feature.

Fig. 10. Video instructions on how to hide the Android Spy app icon from an Android phone [2].



Remove Android's Pesky Cast Icon

With Android Naught (version 7+) and newer they will show an annoying "cast" icon that pcTattletale is running and recording your screen. Follow these steps to remove it:

Fig. 11. Screenshot of tutorial instructions on the pcTattletale app website on how to remove the Pesky Chromecast cast icon for screen recording [1].

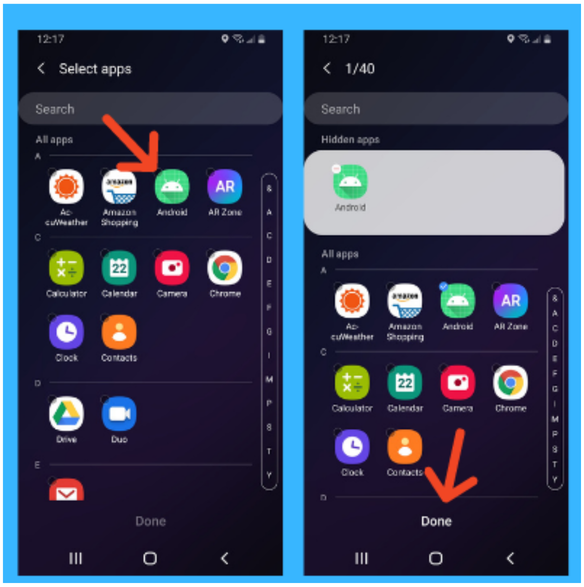


Fig. 12. Screenshot of instructions on how to remove the app pcTattletale app icon from an Android phone (Version 10) [3]. This app has since been removed from the app store.

appears and disappears rapidly when exfiltrating location data, rather than remaining in plain view.

Each app appeared to provide some sort of tracking or information gathering with the exception of com.pctattletale.androidviewer. This app's only role is to guide the user to download an APK hosted outside of the Google Play Store and install it on a Windows, Kindle, or Android device of their choosing. This APK would then capture the screen of the device it is installed on and report it back to the installer. As shown in Figures 11 and 12, the pcTattletale website also provides instructions to remove notifications and hide the app icon.

Lastly, at least 3 apps acted as a keylogger. Of note to these apps were the lack of additional Android permissions requested. One of them (com.as.keylogger) is a free app that redirects the user to another app, Android Spy, for additional functionality such as face capture and GPS tracker. It also provides instructions on how to remove the app icon (see Figure 10). An analysis of dangerous permissions, while useful, nevertheless excludes apps like these.

7 Discussion

7.1 Threats to Validity

We consider the corpus of apps retrieved from the Coalition Against Stalkerware to represent a ground truth. However, some of these apps may potentially represent *dual-use* apps. As discussed in Section 2, we consider their inclusion in our corpus and our analysis of them to be justifiable. The repository of stalkerware apps is curated by leading cybersecurity companies in the industry, who employ malware domain experts with more expertise than our researchers in identifying stalkerware apps. As such we do not attempt to distinguish if there are misidentified dual-use apps in the database. Instead, we recognize their expertise and acknowledge that these apps were added to the threat list based on a set of factors determined by the threat intelligence of the submitting organization.

Further, in Section 6, we use a sample's VirusTotal first submission date in tandem with the app's targetSdkVersion to help determine when the samples of stalkerware were published and active. In both cases, we acknowledge the fact that these metrics are not exact. The VirusTotal first seen date could occur long after the app is first published, and Android supports SDK versions for large periods of time. However, by using the targetSdkVersion to remove noisy samples that were first seen after Google's ToS revisions (but were likely released before, due to having a lower version), we mitigated this issue to the best of our ability.

7.2 Limitations

The keywords we use in Section 5 are primarily English words. However, through manual investigation, we found that some apps and websites were written in other languages (e.g., Russian and Chinese). As such, there is a chance that some apps using monetization schemes were not identified in our analysis.

Furthermore, there is a chance that obfuscation of the apps could play a part in hiding forms of monetization from our keyword searches. To validate our results and identify how many results had been obscured from our keyword search by code obfuscation, we used the peer-reviewed tool LibRadar[7] on the entire app repository, which performs library analysis on the Dalvik bytecode directly [28]. We expanded our initial package name list to match those found by LibRadar, so that we could test if our technique detected the same libraries despite not taking any extra steps to deobfuscate the apps. By adding these extra packages, we were able to identify more apps that used monetization. In doing so we found that our keyword technique was able to identify more ad libraries in 2533 stalkerware apps than LibRadar, and found the same number of ad libraries in 951 instances. Furthermore, the output from our keyword analysis was able to identify every use of the 32 ad libraries that LibRadar was trained on, and identified 25 additional libraries. In both cases, LibRadar was unable to identify ad library uses in apps that our keyword technique was not able to or find instances where ad libraries were used that our keyword search was not able to. As such, we believe our keyword technique was not affected by any obfuscation method that these stalkerware apps may have taken to hide their code.

7.3 Recommendations

From our investigation into the financial ecosystem of stalkerware, we were able to identify key players in each monetization scheme. In Section 4 we showed that Google AdMob is the most popular ad library in both benign and stalkerware apps, however, Google AdMob is the sole ad library drastically more in stalkerware apps than benign apps. In fact, we see that in benign apps it is common to use multiple ad libraries. In contrast, the majority of stalkerware apps use only a single ad library. Furthermore, PayPal and Google AdMob are both the most common financial services in our corpus and the most common monetization scheme to be implemented without any other forms of revenue generation.

In light of our findings, we recommend that both PayPal and Google AdMob moderate the use of their services. The terms of service for Google AdMob specifically prohibits their service to be used for stalkerware.PayPal's Acceptable Use Policy prohibits the use of their payment service in transactions relating to illegal activities or transactions infringing on the right of privacy, both of which apply to stalkerware. Thus these services do not need to change their ToS but simply enforce them better.

We found that Google's change in ToS did not bring the monetization of stalkerware to a halt like Levchenko et al.'s research did in spam emails and pharmaceuticals when they were able to identify a small number of banks where the payments were being processed [27]. Instead, our research shows that stalkerware adapted and changed their monetization schemes. We believe this is a consequence of the decentralized nature of current payment services.

While we recommend increased moderation from PayPal and Google AdMob, we recognize this may not be a long-term solution. Instead we believe that a more sustainable solution may be to invite payment service providers, ad providers, and the different app stores to discuss mitigating stalkerware with stakeholders already involved in these efforts. Direct discussions and collaborations with umbrella efforts such as the Coalition Against Stalkerware could prove to be especially fruitful. By forming a larger group, experts classifying stalkerware apps can directly communicate to the groups officiating the monetization of these apps so that their revenue can be cut off as the apps are identified.

7.4 Ethical Considerations

As we are dealing with a sensitive topic, we have taken steps not to exacerbate the harm caused by these apps. Specifically:

[7] https://github.com/pkumza/LiteRadar

- We used an emulator when analyzing these apps in order to avoid our researchers being harmed by stalkerware.
- We did not subsidize this industry by paying for any apps. Doing so may have revealed further functionality; however, we were unwilling to financially support them.
- We are in the process of disclosing our findings to the relevant parties.
- We did not expose researcher or user data to any of the stalkerware apps we analyzed.
- Any public disclosure, particularly for the general public, will entail anonymization of identifying images and text about these apps.

8 Related Work

Detecting malicious programs and applications is one of the oldest areas of computer security. A wide array of techniques have been proposed, including monitoring filesystem integrity [25], detection of anomalies [14], creating program signatures [26], training machine learning classifiers [37], and more. Unsurprisingly, mobile-specific features and applications (e.g., permissions [34], identifiers [13], etc) has also received a great deal of attention from the research community.

While stalkerware can potentially be detected using many of the techniques proposed for other malicious programs, its threat model is fundamentally different [16]. That is, stalkerware is generally installed by a party with administrative (if not physical) access. Moreover, unlike traditional malware, knowledge by the target that a stalkerware application is installed may be an intentional feature [20]. As such, techniques that detect stalkerware prior to installation and make its monetization difficult are likely necessary to prevent its spread.

As such, characterizing the monetization of stalkerware may be an effective means of combatting it. Nowhere has this approach been more successful than against spam email. For instance, Kanich et al. [23] were the first to characterize message conversion rates, providing realistic estimates of income generated by groups sending these messages. Levchenko et al. [27] took these observations further, identifying that the overwhelming majority of payments to pharmaceutical spam campaigns passed through a small number of processing banks. With this information and in cooperation with these entities, these spam campaigns were shut down virtually overnight. Similar pressure may be possible through the identification of monetization channels for stalkerware.

9 Conclusion

Stalkerware running on mobile platform represents a significant threat to its targets. The ability to report physical location and nearly all of a monitored user's actions allows these applications to facilitate physical and emotional abuse in intimate partner relationships. In this paper, we examine the monetization techniques used by application designers in this space. Through the analysis of over 6,400 applications, we demonstrate not only significant differences in monetization strategies over benign applications (i.e., a lack of revenue from advertisements), but also that the ecosystem is kept funded through a wide range of payment processors. Finally, while policy changes did indeed reduce the number of stalkerware applications using Google's in-app billing, these changes largely pushed such applications to simply employ other payment methods (and in some cases, make no demostrable changes at all). It is our hope that by developing closer relationships between organizations such as the Coalition Against Stalkerware and payment processors that such applications can be made uneconomical in the future.

10 Acknowledgements

We thank the Coalition Against Stalkerware for access to their repository of stalkerware apps, and Avast for providing access to the APKLab.io analysis tool. We also thank William Enck and Samin Yaseer Mahmud for their comments. This work was supported in part by the US National Science Foundation under grant numbers CNS-1562485, CNS-1617474, and CNS-2055123, DARPA HR00118S0052, and Facebook. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily represent the views of the above listed organizations.

References

[1] How to turn off Android?s Pesky Chromecast Icon. https://www.pctattletale.com/blog/3050/how-to-turn-off-androids-

pesky-chromecast-icon/, 2018. Accessed: 2021-08-18.

[2] Hide app symbol. https://www.a-spy.com/en/instructions/hide-launcher/, 2020. Accessed: 2021-08-18.

[3] How To Remove Android 10 Icons. https://www.pcattletale.com/blog/5025/how-to/-remove/-android/-10/-icons/, 2020. Accessed: 2021-08-18.

[4] Coalition Against Stalkerware. https://stopstalkerware.org/, 2021. Accessed: 2021-08-20.

[5] Flash Keylogger. https://play.google.com/store/apps/details?id=tej.flashkeylogger&hl=it&gl=US, 2021. Accessed: 2021-08-18.

[6] uses-sdk. https://developer.android.com/guide/topics/manifest/uses-sdk-element, 2021. Accessed: 2021-08-18.

[7] M. Ahasanuzzaman, S. Hassan, C.-P. Bezemer, and A. E. Hassan. A longitudinal study of popular ad libraries in the google play store. *Empirical Software Engineering*, 25, 01 2020.

[8] Avast. A mobile threat intelligence platform, 2021. Accessed 2021.

[9] R. Bellini, E. Tseng, N. McDonald, R. Greenstadt, D. McCoy, N. Dell, and T. Ristenpart. "So-called privacy breeds evil": Narrative Justifications for Intimate PArtner Surveillance in Online Forums. In *ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*, Dec. 2020.

[10] Brian X. Chen. 'stalkerware' apps are proliferating. protect yourself. https://www.nytimes.com/2021/09/29/technology/personaltech/stalkerware-apps-protection.html, 2021. Accessed November 2021.

[11] R. Chatterjee, P. Doerfler, H. Orgad, S. Havron, J. Palmer, D. Freed, K. Levy, N. Dell, D. McCoy, and T. Ristenpart. The spyware used in intimate partner violence. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 441–458, May 2018.

[12] Coalition Against Stalkerware. What is stalkerware? https://stopstalkerware.org/what-is-stalkerware/, 2021. Accessed May 2021.

[13] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2010.

[14] S. Forrest, S. Hofmeyr, A. Somayaji, and T. A. Longstaff. A Sense of Self for Unix Processes. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1996.

[15] D. Freed, S. Havron, E. Tseng, A. Gallardo, R. Chatterjee, T. Ristenpart, and N. Dell. "Is my phone hacked?" Analyzing Clinical Computer Security Interventions with Survivors of Intimate Partner Violence. In *ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*, Austin, TX, USA, Nov. 2019.

[16] D. Freed, J. Palmer, D. Minchala, K. Levy, T. Ristenpart, and N. Dell. "A Stalker's Paradise": How Intimate Partner Abusers Exploit Technology. In *ACM CHI Conference on Human Factors of in Computing Systems (CHI)*, Montreal, QC, Canada, 2018.

[17] Google. Developer Program Policy: September 16, 2020 announcement. https://support.google.com/googleplay/android-developer/answer/10065487?hl=en, Sept. 2020.

[18] Google. Payments. https://support.google.com/googleplay/android-developer/answer/9858738?hl=en, 2021.

[19] B. Gosangi, H. Park, R. Thomas, R. Gurathi, C. P. Bay, A. S. Raja, S. E. Seltzer, M. C. Balcom, M. L. McDonald, D. P. Orill, M. B. Harris, G. W. Boland, K. Rexrode, and B. Khurana. Exacerbation of Physical Intimate Partner Violence during COVID-19 Lockdown. *Radiology*, 298(1):E38–E45, 2021.

[20] S. Havron, D. Freed, R. Chatterjee, D. McCoy, N. Dell, and T. Ristenpart. Clinical computer security for victims of intimate partner violence. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security 19)*, pages 105–122, Santa Clara, CA, August 2019. USENIX Association.

[21] C. Heasley. Android Stalkerware. https://github.com/diskurse/android-stalkerware, 2020. Accessed August 2021.

[22] M. Iqbal. App download and usage statistics (2020). https://www.businessofapps.com/data/app-statistics/, 2021. Accessed July 2021.

[23] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage. Spamalytics: An empirical analysis of spam marketing conversion. *Commun. ACM*, 52:99–107, 09 2009.

[24] C. Khoo, K. Robertson, and R. Deibert. *Installing Fear: A Canadian Legal and Policy Analysis of Using, Developing, and Selling Smartphone Spyware and Stalkerware Applications*. The Citizen Lab Research Report No. 120, University of Toronto, June 2019.

[25] G. Kim and E. H. Spafford. The Design and Implementation of Tripwire: A File System Integrity Checker. In *Proceedings of the ACM Conference on Computer and Communications Security*, 1994.

[26] S. Kumar and E. Spafford. A Generic Virus Scanner for C++. In *Proceedings of the Computer Security Applications Conference*, 1992.

[27] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Felegyhazi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, D. McCoy, N. Weaver, V. Paxson, G. M. Voelker, and S. Savage. Click Trajectories: End-to-End Analysis of the Spam Value Chain. In *IEEE Symposium on Security and Privacy*, 2011.

[28] Z. Ma, H. Wang, Y. Guo, and X. 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, New York, NY, USA, 2016. Association for Computing Machinery.

[29] S. Y. Mahmud, A. Acharya, B. Andow, W. Enck, and B. Reaves. Cardpliance: PCI DSS compliance of android applications. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1517–1533. USENIX Association, Aug. 2020.

[30] Malwarebytes Labs. Phone screenshots accidentally leaked online by stalkerware-type company. https://blog.malwarebytes.com/stalkerware/2021/09/phone-screenshots-accidentally-leaked-online-by-stalkerware-company/, 2021. Accessed November 2021.

[31] National Network to End Domestic Violence. A glimpse from the field: How abusers are misusing technology. *Safety Net Technology Safety Survey 2014*, 2014.

[32] C. Parsons, A. Molnar, J. Dalek, J. Knockel, M. Kenyon, bennett Haselton, C. Khoo, and R. Deibert. *The Predator in Your Pocket: A Multidisciplinary Assessment of the Stalkerware Application Industry*. The Citizen Lab Research Report No. 119, University of Toronto, June 2019.

[33] PNF Software. JEB Decompiler by PNF Software. https://www.pnfsoftware.com/, 2022. Accessed: March 14th, 2022 [Online].

[34] A. Porter Felt, E. Chin, S. Hanna, D. Song, and D. Wagner. Android Permissions Demystified. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2011.

[35] K. A. Roundy, P. B. Mendelberg, N. Dell, D. McCoy, D. N. Nissani, T. Ristenpart, and A. Tamersoy. The many kinds of creepware used for interpersonal attacks. In *IEEE Symposium on Security and Privacy (SP)*, 2019.

[36] Skylot. GitHub skylot/jadx - Dex to Java decompiler. https://github.com/skylot/jadx, 2022. Accessed: March 14th, 2022 [Online].

[37] R. Sommer and V. Paxson. Outside the Closed World: On Using Machine Learning For Network Intrusion Detection. In *Proceedings of IEEE Symposium on Security and Privacy*, 2010.

[38] L. Stefanko. Android Stalkerware Vulnerabiltiies. Technical report, ESET, May 2021.

[39] K. Thomas, D. Akhawe, M. Bailey, D. Boneh, E. Burzstein, S. Consolvo, N. Dell, Z. Durumeric, P. G. Kelley, D. Kumar, D. McCoy, S. Meiklejohn, T. Ristenpart, and G. Stringhini. SoK: Hate, Harassment, and the Changing Landscape of Online Abuse. In *IEEE Symposium on Security and Privacy*, May 2021.

[40] E. Tseng, R. Bellini, N. McDonald, M. Danos, R. Greenstadt, D. McCoy, N. Dell, and T. Ristenpart. The Tools and Tactics Used in Intimate Partner Surveillance: An Analysis of Online Infidelity Forums. In *USENIX Security Symposium*, 2020.

[41] E. Tseng, D. Freed, K. Engel, T. Ristenpart, and N. Dell. A Digital Safety Dilemma: Analysis of Computer-Mediated Computer Security Interventions for Intimate Partner Violence During COVID-19. In *ACM CHI Conference on Human Factors of in Computing Systems (CHI)*, Yokohama, Japan, May 2021.

[42] G. Venkatadri, P. Sapiezynski, E. M. Redmiles, A. Mislove, O. Goga, M. Mazurek, and K. P. Gummadi. Auditing offline data brokers via facebook's advertising platform. In *The World Wide Web Conference*, WWW '19, page 1920–1930, New York, NY, USA, 2019. Association for Computing Machinery.

[43] World Health Organization. Violence against women. https://www.who.int/en/news-room/fact-sheets/detail/violence-against-women, 2021. Accessed May 2021.