2018 IEEE Symposium on Security and Privacy

The Spyware Used in Intimate Partner Violence

Rahul Chatterjee*, Periwinkle Doerfler†, Hadas Orgad‡, Sam Havron§, Jackeline Palmer¶, Diana Freed*, Karen Levy§, Nicola Dell*, Damon McCoy†, Thomas Ristenpart*

* Cornell Tech † New York University ‡ Technion § Cornell University ¶ Hunter College

***Abstract*—Survivors of intimate partner violence increasingly report that abusers install spyware on devices to track their location, monitor communications, and cause emotional and physical harm. To date there has been only cursory investigation into the spyware used in such intimate partner surveillance (IPS). We provide the first in-depth study of the IPS spyware ecosystem. We design, implement, and evaluate a measurement pipeline that combines web and app store crawling with machine learning to find and label apps that are potentially dangerous in IPS contexts. Ultimately we identify several hundred such IPS-relevant apps.**

While we find dozens of overt spyware tools, the majority are "dual-use" apps — they have a legitimate purpose (e.g., child safety or anti-theft), but are easily and effectively repurposed for spying on a partner. We document that a wealth of online resources are available to educate abusers about exploiting apps for IPS. We also show how some dual-use app developers are encouraging their use in IPS via advertisements, blogs, and customer support services. We analyze existing anti-virus and anti-spyware tools, which universally fail to identify dual-use apps as a threat.

I. Introduction

Intimate partner violence (IPV) affects roughly one-third of all women and one-sixth of all men in the United States [54]. Increasingly, digital technologies play a key role in IPV situations, as abusers exploit them to exert control over their victims. Among the most alarming tools used in IPS are spyware apps, which abusers install on survivors' phones in order to surreptitiously monitor their communications, location, and other data. IPV survivors [23, 29, 46], the professionals who assist them [29, 58], and the media [9, 22, 37] report that spyware is a growing threat to the security and safety of survivors. In the most extreme cases, intimate partner surveillance (IPS) can lead to physical confrontation, violence, and even murder [10, 18].

The definition of "spyware" is a murky one. Some apps are overtly branded for surreptitious monitoring, like FlexiSpy [2] and mSpy [6]. But survivors and professionals report that other seemingly benign apps, such as family tracking or "Find My Friends" apps [8, 29, 58], are being actively exploited by abusers to perform IPS. We call these *dual-use apps*: they are designed for some legitimate use case(s), but can also be repurposed by an abuser for IPS because their functionality enables another person remote access to a device's sensors or data, without the user of the device's knowledge. Both overt spyware and dual-use apps are dangerous in IPV contexts.

We provide the first detailed measurement study of mobile apps usable for IPS. For (potential) victims of IPS, our results are decidedly depressing. We therefore also discuss a variety of directions for future work.

Finding IPS spyware. We hypothesize that most abusers find spyware by searching the web or application stores (mainly, Google Play Store or Apple's App Store). We therefore started by performing a semi-manual crawl of Google search results. We searched for a small set of terms (e.g., "track my girlfriend's phone without them knowing"). In addition to the results, we collected Google's suggestions for similar searches to seed further searches. The cumulative results (over 27,000+ returned URLs) reveal a wide variety of resources aimed at helping people engage in IPS: blogs reviewing different apps, how-to guides, and news articles about spyware. We found 23 functional apps not available on any official app store, and a large number of links to apps available on official app stores.

We therefore design, build, and evaluate a crawling pipeline for Google Play [3], the official app marketplace for Android. Our pipeline first gathers a large list of potential IPV-related search terms by using search recommendations from Play Store, as we did with Google search. We then collect the top fifty apps returned for each of the terms. Over a one-month period, this approach retrieved more than 10,000 apps, though many have no potential IPS use (e.g., game cheat codes were returned for the search term "cheat").

The data set is large enough that manual investigation is prohibitive, so we build a pruning algorithm that uses supervised machine learning trained on 1,000 hand-labeled apps to accurately filter out irrelevant apps based on the app's description and the permissions requested by the app. On a separate set of 200 manually labeled test apps, our classifier achieves a false positive rate of 8% and false negative rate of 6%. While we do not think this represents sufficient accuracy for a standalone detection tool given the safety risks that false negatives represent in this context, it suffices for our measurement study. We discuss how one might tune the pipeline to incorporate manual review to achieve higher accuracy (and no false negatives), as well as initial experiments with crowdsourcing to scale manual review.

We performed a smaller study using our measurement pipeline with Apple's App Store, and got qualitatively similar results. See Appendix B.

The IPS landscape. The resulting corpus of apps is large, with hundreds of Play Store applications capable of facilitating IPS. We manually investigate in detail a representative subset of 61 on-store and 9 off-store apps by installing them on research phones, analyzing the features and user interface they

DOI 10.1109/SP.2018.00061

provide, and observing how they are marketed. We uncover three broad categories of apps: personal tracking (e.g., find-my-phone apps), mutual tracking (e.g., family tracking apps), and subordinate tracking (e.g., child monitoring apps).

The three types of apps have differing capabilities, though all can be dangerous in an IPS context. The worst allow covert monitoring of all communications, remote activation of cameras and microphones, location tracking, and more. Two of the on-store apps we analyzed, Cerberus and TrackView, violate Play Store policy by hiding their app icon and showing no notifications, making them as covert as off-store spyware. (We reported these apps to Google for review, see discussion about our disclosures below.) All 70 apps are straightforward to install and configure, making them easy to use by abusers.

Some off-store apps overtly advertised themselves for use in IPS. An example is HelloSpy, whose website depicts a man physically assaulting a woman with surrounding text discussing the importance of tracking one's partner, see Figure 1. Others, including those on the Play Store, most often do not have descriptions or webpages promoting IPS. However, further investigation revealed that a number of these apps advertised or condoned IPS as a use case. We document that vendors advertise on IPS-related search terms such as "how to catch cheating girlfriend" on both Google and Play Store. We also uncover networks of IPS-focused websites that link exclusively to a specific app's webpage and directly advertise IPS use cases for the app.

For a subset of 11 apps (6 on-store and 5 off-store), we contacted customer service representatives posing as a potential abuser.[1] In response to the question "If I use your app to track my husband will he know that I am tracking him?", 8 out of 11 responded with affirmative explanations implicitly condoning IPS. Only one (an off-store app) replied with an admonishment against use for IPS. Two apps did not respond.

Performance of anti-spyware tools. The existence of so many easy-to-use, powerful apps usable for IPS demonstrates that victims need detection and cleanup tools. A variety of tools advertise their ability to deal with spyware. These include tools from major anti-virus vendors, such as Symantec, Kaspersky, and Avast, as well as some lesser-known tools. As far as we are aware, no one has evaluated any of these tools for the particular task of detecting IPS spyware or dual-use apps. We evaluate anti-spyware tools against a corpus of 280 on-store apps detected by our crawl of Google Play (that we manually verified to be usable for IPS) and all 23 off-store spyware apps we identified.

No anti-spyware tool effectively detects IPS-relevant apps. The best performing (Anti Spy Mobile) flagged 95% of off-store spyware, but only 47% of on-store IPS-relevant apps. The tool also has a prohibitively high false positive rate of 12%, labeling applications such as Google Chrome and Play Store as spyware. The major anti-virus systems were some of the worst performers for dual-use apps (flagging at most 13% of on-store apps). While this labeling may be appropriate for other contexts, for IPV victims these tools are far too conservative. The tools do better detecting off-store spyware, but still fail to label some dangerous apps.

[1]Our IRB board confirmed that our experiments are exempt from review, as they engage people in their professional capacities and do not collect PII.

Fig. 1: Screenshot of the HelloSpy website promoting intimate partner violence [35].

Summary and next steps. We perform the first study of applications usable for IPS. In particular:

- We introduce measurement approaches for discovering applications easily found by abusers via searching the web and app stores. The discovered apps pose immediate and dangerous threats to victims.
- We highlight the role of dual-use apps in IPS, and show that they are often as powerful as overt spyware. On-store apps can achieve prohibited capabilities due to a lack of OS-level protections.
- We show that many apps brand themselves only for "legitimate" purposes, but simultaneously pay for IPS-related advertisements. A small measurement study revealed that some apps' customer service representatives condone IPS.
- We show that existing anti-virus and anti-spyware tools are ineffective at detecting and remediating IPS spyware.

While our first-of-its-kind study has various limitations (see Section III-D), it nevertheless uncovers the prevalence of apps that can facilitate IPS, and which can lead to immense emotional, psychological, and physical harm for victims.

We disclosed our results to Google, and they have already taken steps to improve safety for their users. Google reviewed the apps we discovered and confirmed that they took action against some of the apps that violated Play Store policies due to a lack of persistent notifications or promotion of spyware or stealth tracking. In addition, Google is expanding their restrictions on advertisement serving for IPV-related queries.

We hope our results will motivate the computer security community more broadly to work to improve survivor safety. We therefore conclude with an initial discussion about next steps, including how to: deal with the complexities of dual-use apps; improve detection tools for use in IPV settings; suggest ways honest developers can prevent exploitation of their tools for IPS; and modify laws or regulations to better help survivors.

II. Background: Spyware in IPV

Intimate partner violence (IPV) includes physical or sexual violence, stalking, or psychological harm by a current or former intimate partner or spouse. A number of studies [23,29, 40,58] indicate that abusers increasingly exploit technology to monitor and control their partner in IPV contexts, which can be a form of abuse in and of itself and can facilitate other forms of abuse (physical, emotional, sexual, etc.).

Much of the IPV literature discusses the installation of IPS apps on a survivors' mobile devices [23, 28, 29, 40, 46, 58]. One study [46] interviewed 15 survivors of IPV in the United States, and found that 20% reported being monitored by spyware. An analysis of data stolen from two spyware vendors, FlexiSpy and Retina-X, revealed that 130,000 people use one of the tools [22]. The hackers' investigation concluded that most usage is for IPS. Interviews of survivors and professionals working with them indicate that abusers can easily find spyware via web search, and that many otherwise innocuous apps, such as "Find my phone" apps and child trackers, are easily repurposed by abusers for spying on intimate partners [23,29,40,53,58].

Spyware or other apps that facilitate surveillance are particularly dangerous in IPV situations because abusers often have physical access to their partner's device(s), and can know, guess, or compel disclosure of access credentials (passwords, PIN codes, or swipe patterns) [29, 46, 58]. This enables the abuser to install spyware via an app store such as the Google Play Store or Apple App Store. In the case of Android, an abuser can configure the phone to allow installation of apps not found on the Play Store. Installation of apps does not require sophisticated technical knowledge and, as we discuss in detail later, easy-to-follow installation guides are readily available. All of the spyware we encountered can be used without rooting the phone, though in some cases additional spying features were available should one do so.

Types of IPS apps. Our focus is on apps that abusers purposefully install in order to stalk, monitor, and control an intimate partner's device without their consent. The examples above indicate two main classes of spyware. We refer to apps like "Find my phone" as *dual-use apps*, as they can be deployed as spyware despite not being purposefully designed for such use. In contrast, *overt spyware* like FlexiSpy and Retina-X are designed and advertised to be surreptitious and applicable for spying on a target.

We will use the term *IPS-relevant apps* or *IPS apps* to refer to apps that we believe may be purposefully installed by abusers for surveillance; this category includes both overt spyware and dual-use apps. In more detail we consider as IPS-relevant apps those (1) whose primary purpose is giving another person the ability to collect data, track location, and/or remotely control a device; (2) which function, after initial installation and configuration, without interaction with the current user of the device; and (3) that the victim most likely does not want on the device. This means we will, in general, not consider apps such as Google Maps: while it can be configured to continuously update another person of the device's location without interaction with the current user, its primary purpose is not to enable location tracking by another person and most victims want it installed. Certainly such apps have safety and privacy implications in IPV, but, particularly because victims might desire their continued installation, their analysis and remediation will require different approaches.

These scoping rules will not always be easy to apply in a decisive way. When in doubt we conservatively marked apps as IPS relevant. In most such cases we anyway found anecdotal evidence online of abusers employing the app or similar ones.

Other forms of malware. We do not consider adware (sometimes called commercial spyware) or other potentially unwanted programs (PUPs), that help companies collect information on user behavior. The ecosystem of PUPs has been analyzed in [38, 56]. Moreover, we don't specifically consider advanced malware such as those used by governments or the remote access trojans (RATs) [25, 39, 43] often used by voyeurs, which generally would require more technical sophistication on the part of an abuser to deploy for IPS. Likewise there have been many prior studies measuring and detecting more commercially-motivated malware that steal users secrets (e.g., bank details) [12, 14, 26, 34, 60, 62]. While in theory it could be that some spyware or anti-spyware apps double as malware that aims to leak confidential information to third-parties (people other than the abuser), we have found no evidence of intention[2] by vendors to do so. These other forms of unwanted software certainly carry privacy risks in IPV contexts as well as elsewhere, but their study and remediation require different techniques than those we explore here.

Open questions. Despite the many indications that spyware is widely used in IPV, there has been, to date, no in-depth study of the technologies available to abusers. Thus, our work endeavors to answer a number of critical open questions:

- How easy is it to find apps usable for IPS? How many such apps are available?
- Can we find and categorize the kinds of dual-use apps that might be used in IPS?
- What capabilities are available to abusers?
- Are app developers encouraging IPS?
- Are there effective tools (e.g., anti-spyware) for detecting and removing such apps?

III. Finding IPS-Relevant Apps

In this section, we perform measurements to discover apps usable for IPS. We focus on the apps that an abuser, assumed to be of average technical sophistication, could locate and deploy. To this end, we ignore apps that are difficult to locate (e.g., advertised in closed forums) or difficult to deploy (e.g., require rooting a phone). We instead look at apps that can be readily

[2]Some spyware apps do have vulnerabilities that accidentally leak data to third-parties, see Section IV.

found by searching either in a popular search engine such as Google or in an official app store.

We emulate a hypothetical abuser seeking apps for IPS. We hypothesize that most abusers begin by performing searches such as "track my wife" or "read SMS from another phone" in a search engine. Under this hypothesis, we gather examples of both resources for abusers (such as how-to guides) and apps readily found by abusers. While our measurement methodology may not uncover all IPS apps, we believe it surfaces a representative sample of apps as used by abusers because: (1) our approach uncovers a huge number of IPS tools that (2) cover in aggregate all the types of tools reported by survivors [16, 29, 40].

Below we describe our methodology for searching in greater detail. For concreteness and because of its large market share, we focus on the Android ecosystem, specifically using search interfaces provided by Google.com and Google Play. Our techniques can be applied to other ecosystems that provide a search engine; see Appendix B for our treatment of Apple.

Methodology. To perform searches, we need a wide-ranging list of queries that an abuser might use. For this, we utilize *query recommendation* APIs provided by search engines, such as a query completion API (provided by Google Play) and a related query API (obtained by parsing the search results page of Google). A query completion API responds with a number of recommended search phrases that contain the submitted query as a substring, whereas a related query API responds with a set of search phrases believed to be semantically related to the submission. In both cases the intent of the APIs is to suggest related search phrases as educated by prior searches made by other users of the engine.

We use a *query snowballing* approach to find a large set of useful queries given a small set of queries as a seed set. The procedure is straightforward: query the recommendation APIs on all queries in the seed set, collect the resulting recommendations, query the resulting recommended search terms, and continue until some predetermined number of queries have been discovered (e.g., $\ell = 10{,}000$ search phrases), or until we converge to a set where no new recommendations are found. More on query snowballing is given in Appendix A.

A. Searching for IPS on Google

We begin by applying our query snowballing to the Google search engine. We used the Python Requests library [17] to make the queries and download the results, and the Lxml library [5] to parse the pages. Search results vary based on, among many other factors, the query browser and the search history. We used a user-agent identifying the request as from a Chrome browser on Linux and disallowed any client-side cookies to minimize influence of historical searches.

Google's related query suggestions provide semantically close queries, so we use in our seed set relatively long and complete queries, as opposed to the smaller seed terms we use for Google Play (see below). We begin with queries such as, "how to catch my cheating spouse" or "track my husband's phone without them knowing." The returned suggestions are not always relevant to our study, e.g., suggestions for "cheat" include suggestions related to cheat codes for video games. We therefore filter the queries using regular expression blacklists (built via manual inspection). The initial set of seed queries and the blacklists used are given in Appendix A.

Type	Description	#	Example
Blogs	How-to blogs for IPS	21	best-mobile-spy.com
Videos	How-to videos for IPS	12	youtube.com
Forums	Q&A forums for IPS	7	quora.com
News	News about using spying software	2	theguardian.com
Downloads	Pages hosting IPS apps	12	download.cnet.com
App sites	Websites of apps	5	thetruthspy.com
App store	Link to apps in the official app stores	2	GPS Phone Tracker
Other	Irrelevant pages	39	amazon.com

Fig. 2: Types of websites found in manual analysis of 100 randomly sampled URLs from our 27,741 URLs found via Google search.

Our snowballing process did not converge even after considering query sets of sizes up to $\ell = 10{,}000$, therefore we consider all 10,000 query recommendations. We submitted each of these queries to Google and recorded the top 10 results for each query. From these searches, we collected 27,741 URLs on 7,167 unique domains. We manually investigated a random subset of 100 URLs to group their associated websites into six major categories; see Figure 2. Nearly two-thirds of the sampled pages are directly related to IPS, with only 39 URLs linking to unrelated content. We now discuss the 61 IPS-related URLs, first those that provide information about how to engage in IPS, and then those that link to IPS apps.

Information about conducting IPS. The majority of the IPS-related URLs (65%) link to blogs, videos, or question-and-answer forums discussing how to engage in IPS. The blogs describe how to use one or more tools to spy on someone. Example blog post topics include "Read your wife's messages without touching her phone" on a blog linking to mSpy and "These apps can help you catch a cheating spouse" appearing on the NY Post news site. News articles that came up in our searches also point to incidents of spyware being used for IPS. All these serve to direct those wanting to engage in IPS to app websites, even should that website try to distance itself from IPS. We discuss more about disingenuous blogging in Section V. The video tutorials (mostly hosted on Youtube) similarly explain how to setup and use apps for spying.

The question-and-answer forums focus on spying or tracking with discussions about how to use various tools for spying on intimate partners. For example, in one forum an (ab)user posts *"I'm looking for an app I can install on my wife's phone that is hidden so that I can see where she is or has been via cell towers or gps."* In reply, another (ab)user posts,

> *[Install] Cerberus from the market. Once installed and configured, can be set to be invisible in the app drawer. You can also record audio and take pictures remotely with it! Be sure to silence the camera first though!*

IPS apps. The remaining IPS-related URLs linked to home pages for apps, links to Google Play application pages, or

websites that aggregate a number of download links for apps. From the Google search results and in the resulting web pages, we collected 2,249 unique URLs pointing to Google Play (extracted using regular expression search), among these URLs we found 1,629 active apps listed in Google Play. Manual analysis of a random sample of 100 of these apps revealed that 22 were usable for IPS. All of which were separately discovered by our search in the Play store (see Section III-B). The prevalence of Google Play links found via search on Google suggests that on-store apps will be found by abusers. The dual-use nature of most of these on-store apps suggests that tools used for IPS are, and will continue to be, allowed on app stores.

To gather actual examples of off-store apps (distributed outside Google Play), we examined all references to apps found in the 100 manually-analyzed URLs. We also found 479 domains (among the full set of 7,167 domains returned by searches) containing the words "spy", "track" or "keylog". We investigated a random subset of 50 such domains, and found that either they are discussing or hosting spyware apps. Finally, we came across a web service called AlternativeTo [11], which gives suggestions for alternative apps for some queried application. We queried this service with the apps we had identified thus far to find more spyware apps.

Ultimately we found 32 unique off-store apps. These all constitute overt spyware, as they advertise their ability to surreptitiously track and monitor a device. Nine of the apps had been discontinued at the time of our study and are no longer available. The remaining 23 serve, in later sections, as our corpus of off-store apps. We believe this corpus comprehensively represents a current snapshot of off-store spyware: in many subsequent manual searches about IPS related topics in the course of this research, we did not find any reference to additional off-store spyware.

B. Searching for IPS-relevant apps in Google Play

Our results above revealed that apps on Google Play come up when searching Google for IPS-related phrases. We therefore investigate Play Store directly, to see what types of IPS-related tools it hosts.

We perform a similar query snowballing procedure as discussed above using the query completion API provided by Google Play with smaller seed queries (as opposed to the longer ones used in the previous Google search, see Appendix A). In Google Play search, the snowball querying converged rapidly to a final set of suggested phrases.

With each phrase in the final set, we search Google Play and collect the metadata of the first 50 apps returned. This metadata contains, among other information, the description of the app, the minimum version of Android supported by the app, the date of the last update to the app in the Play Store, a range for the number of downloads, the average user rating, and a unique identifier (called the Android app ID). For each application we also downloaded the most recent reviews (up to 200, the API limit), and the requested permissions as listed in the application's manifest file. We did this search using a modified version of an unofficial scraper for Google Play called google-play-scraper [48]. We limit our searches to at most five queries per second to minimize any operational overhead on the search engines. This type of scraping is generally considered acceptable research behavior [45].

Every day[3] for one month starting on Oct 23, 2017, we repeat query snowballing, and then perform searches on the cumulative set of queries found in earlier days.

Results. On average the size of the query snowball retrieved each day was 530 (with a standard deviation of 6). The set of queries retrieved every day changed over time even though the seed queries were the same. Every week, we saw about 40 new queries gathered using our snowballing approach. The total size of our query pool was 675 after one month of crawling. In Figure 3a we show the change in the number of terms we saw via snowballing and the total size of the query pool each day. Google updates their query completion API periodically to incorporate recent searches by users, which might account for the periodic increase in the cumulative set of terms even after crawling for a month.

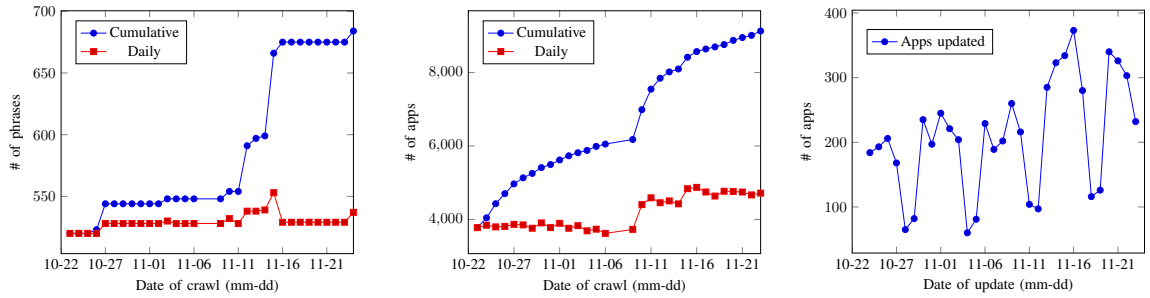
The number of apps found in this process also varies over time — rather more rapidly than the queries. We found an average of 4,205 unique apps each day (with standard deviation 450). We saw on average 288 new apps each day, with a similar number of apps going missing (ones found in previous days not found via our search procedure on a given day). See Figure 3b. In total we collected 9,224 apps.

Our measurement suggests that the results of searches change significantly over time. Part of this is the change in the set of queries searched, and the rest is due to the fact that many apps are removed from Google Play, some are updated, and new apps are posted. Looking at the update date, we found that 32% of all 9,224 apps were updated at least once, and 15% were updated three times during our one month of study. Every week on average 15% of the apps' binaries were updated, with the highest number of apps being updated on Mondays and Thursdays (see Figure 3c). We also found that 208 (2%) of the observed apps were removed from the Google Play store (the Google Play pages of these apps return HTTP error 404) during our study period. Apps may have been removed by Google or by the developer, though we do not know which is the case for these apps.

Developers can classify their apps within a fixed set of genres, which improves discovery of the app. However, we found inconsistencies in the reported genres of some apps. For example, an IPS-relevant app titled Friends & Family Tracker was listed as a casual gaming app.

Among the 9,224 searched apps, many were not relevant to our study. For example, the search results include many apps for tracking finances or pregnancy, which cannot be used as spyware. This necessitated a mechanism for pruning apps that are not IPS-relevant.

[3]Scans were not performed on Nov 07 and Nov 08 due to a power failure.



(a) (b) (c)

Fig. 3: (a) The size of Google Play recommendation snowballs each day, and the size of the cumulative set of distinct queries. (b) The number of distinct apps found each day, and the cumulative number of apps over time. (c) Number of apps updated each day by the developer.

C. *Pruning false positives*

Many of the apps we discovered via Play Store search are not relevant to IPS, as per our scoping discussed in Section II. We therefore need to filter out such false positives. The large number of apps suggests the need for a scalable approach.

Pruning via machine learning. We decide to use supervised machine learning to help filter out apps that are not IPS-relevant. We hand labeled 1,000 randomly sampled apps from the 3,777 apps we found in the first day of crawling (ignoring the apps whose descriptions are not in English); we refer to this dataset as TR. We labeled them as IPS tools or benign based on the information available on their Google Play page. Of these 1,000 apps 280 (28%) were marked as IPS tools.

In building the model, we consider the description, summary, genre, and list of required permissions of the apps. We tried other information, such as installation count and reviews, but did not find any improvement in accuracy.

We used a bag-of-words (BoW) model for the descriptions and summaries using the CountVectorizer function provided by the Scikit-Learn [50] library. We considered n-grams of words, for $1 \leq n \leq 4$ to construct the BoW model, ignoring those that appear in less than 1% of the apps or more than 95% of the apps. From the BoW model we picked the 1,000 most discriminatory features based on the χ^2-statistic [61]. We treated each permission and each genre as individual words, and created a BoW model for them. We picked the 50 most discriminatory features from this model based on χ^2-statistic. Finally, we took the union of these features to represent each app in a 1,050-dimensional feature space.

To train our model, we tried different machine learning approaches, which we compared using an area under the curve (AUC) metric [33]. For each ML algorithm, we perform 10-fold cross validation with randomly selected folds using the hand-labeled training data TR, and then considered the average value of AUC across all folds. We used Python Scikit-Learn [50] to train and evaluate machine learning models.

We found that logistic regression (LR) with L_2 penalty and inverse of the regularization strength (C) set to 0.02094 (found via grid search) worked the best, giving an AUC value of 0.94 (optimal value is 1.0). This leads to a false positive rate (FPR) of 4% and a false negative rate (FNR) of 4%. We tested, among other algorithms, decision trees, random forests, K-means, and SVM, and found none performed better than LR.

Threshold		TR	TS_1	TS_2	TS_{1+2}
0.5	Accuracy	96	91	95	93
	FNR	4	4	10	6
	FPR	4	11	6	9
0.3	Accuracy	86	82	81	82
	FNR	< 1	0	0	0
	FPR	19	25	24	25

Fig. 4: Performance (in percent) of LR classifier on training and different test sets for two classification thresholds.

Evaluation. We finally evaluate our machine learning model on 200 apps from two different time periods. Half of these apps (denoted by TS_1 hereafter) are sampled from the first week's 6,361 apps (omitting the first day's results that were used to select the 1,000 training apps) and the other half (denoted by TS_2) were sampled from the fourth (last) week's 7,581 apps. We hand-label the 200 apps as benign or IPS-relevant as before. TS_1 has 28 IPS-relevant apps, while TS_2 has 22.

In Figure 4 (first group of rows, with cutoff 0.5) we note the accuracy, FPR, and FNR of the logistic regression model on the training data (TR) and the two test sets (TS_1 and TS_2). We see that the LR classifier generalizes well, as the test accuracy is close to that of the training dataset. Moreover, the model handles concept drift well: apps from a month later are as accurately classified as those coming from the same time period. Averaging across the entire test set (TS_{1+2}), the classifier achieves 93% accuracy with 6% false negatives.

We would like to minimize false negative rates — erroneously classifying an app usable for IPS as benign. Looking ahead to potential use of our classifier as a detection tool, failing to detect IPS apps on a phone is dangerous in many IPV settings; while misclassifying benign apps creates overhead, but is relatively harmless. We thus experimented with multiple classification thresholds (how confident does the LR model need to be before we classify something as IPS-relevant). We found that a threshold of 0.3 (as opposed to a standard 0.5; the positive class is IPS apps) achieves false negative rate below 1% and false positive rate at 19%, with 34% of all apps marked as relevant. These numbers are averages over 10

random folds of the training data. The performance at this threshold on test data appears in Figure 4.

The false positive rate could be reduced via manual inspection of the ML-pruned apps. For example, in subsequent sections we only investigate apps that we manually verified to be IPS-relevant. Towards scaling manual inspection, we explored using Amazon Mechanical Turk, see Appendix C.

D. Limitations of our app discovery approach

There are a few limitations to our app discovery approach. First and foremost, we only focused on English-language search queries and on apps with descriptions in English. Therefore, some spyware used in non-English-speaking communities may be missed. That said, our methods can readily be localized to other languages.

Our initial seed queries are manually picked and the snowballs do not represent an exhaustive set of search terms that an abuser might use. As a result, it could be that our techniques missed some IPS apps.

Our machine learning and manual labeling approaches primarily relied on descriptions on the Google Play store, but some apps have only cursory, vague, or incomplete descriptions. Some apps have capabilities not listed in their description. Other apps promise capabilities they do not deliver. (In the next section we discuss some examples.) Many of the apps falling into this category have more comprehensive specifications on a separate website, and future work might attempt to additionally leverage this information to improve accuracy. Likewise, using natural language processing techniques (e.g., [49]) might help in improving accuracy. As another route to improvements, one might augment our techniques with direct analysis of app binaries, perhaps using the rich set of techniques that have been developed to analyze (other kinds of) malware apps [14, 24, 32, 44, 59].

Finally, what exactly should be considered IPS-relevant is not always clear, even to expert human analysts. Our ground truth labels may therefore contain some errors for apps on the margin, and we tended to bias towards conservatively marking apps as IPS-relevant, for the same reasons we tuned our classifier towards a low false negative rate. This viewpoint seems appropriate, given the many online resources we found that suggest using truly well-intentioned apps (such as folder synchronization tools) for IPS.

IV. IPS-Relevant App UX and Capabilities

In Section III we discuss how we discovered IPS tools through manual and automated crawling. Here we dig into the types of apps found. We group them into various high level categories, and then analyze both their user experience (from the perspective of both abusers and victims) as well as their capabilities.

App selection. We manually investigated 70 apps chosen from our corpus of apps: 61 from Google Play (on-store), and 9 from the open web (off-store). The apps were selected as follows: We ordered on-store apps in decreasing order of their download counts and chose apps until we had at least three apps from each category (see Figure 5). We capped the maximum number of on-store apps to consider for a category to 15, ignoring apps with lower download counts. Of the 23 off-store apps we observed, 18 apps could be downloaded without entering any credit-card information, whereas the remaining 5 needed to be purchased. We randomly selected 6 of the free apps and 3 that required purchase.

For each app, a researcher reviewed the description of the app, installed it on a simulated victim phone, installed any complementary version on a simulated attacker phone (both phones running Android 6.0), and recorded the capabilities provided by the app. We found that 12 of the 70 apps were buggy or did not work in accordance with their description; they are excluded from the discussion below.

We observed that most apps fell into three categories based on their intended usage.

- **Personal tracking**: These are apps intended for use solely by the owner of a phone. Examples include text message forwarding services and anti-theft (Find-my-phone) apps.
- **Mutual tracking**: These apps allow a group of people to track each other's locations. Examples include Find-my-family apps, or couple trackers.[4]
- **Subordinate tracking**: These apps are designed to enable one party to track another, and not vice versa. Examples include child or employee monitoring apps. Most off-store IPS spyware falls into this category.

In Figure 5, we summarize these categories, with examples.

Some of the on-store apps that we investigated seemed, in our assessment, to violate Play Store policy. We therefore reported them to Google, who subsequently reviewed the apps and took action against all those apps that they found to violate their policy. These included some that lacked a persistent notification or that promoted themselves as spyware or stealth tracking (see discussion below).

A. (Ab)user experience

Assuming physical access to a victim's unlocked device, installation and configuration of most apps is straightforward. Prior work [40, 58] reports that abusers often have access to victims' phones and either know, can guess, or can compel disclosure of the credentials needed to unlock it.

Most of the apps we evaluated, both on and off Play Store, have a subscription payment model with tiered pricing for a range of capabilities. Some have free trials or free versions with limited capabilities. The popular dual-use apps (with more than 10 million downloads) on Play Store cost somewhere between $5 for a lifetime upgrade (Wheres My Droid) to $10 USD per month (TrackView). In contrast, the apps that are distributed on the web range in cost from $20 to $50 USD per month (for up to five phones).

On-store apps can be installed via the Play Store app on the device. To install off-store apps, the abuser must first configure the device to allow installation of apps from "unknown

[4]Couple trackers' benign use case is for consensual location and information sharing between partners, differentiating it from their dual-use in IPS.

	App types	Description	Examples	Capabilities
Personal tracking	Find-my-phone	Locate phone remotely	Find my Android	Location tracking, remote locking and wiping
	Anti-theft	Catch the phone thief	Wheres My Droid	Record location, photos & ambient audio; alert on SIM change
	Call recorder	Record incoming / outgoing calls	Call Recorder	Record calls and back them up to a server
	Data syncing	Sync data from phone to other device	mySMS	Sync SMS and call log, media, browser history
	Phone control	Control phone remotely	TrackView	Full control with capabilities exceeding combination of data syncing and anti-theft
Mutual tracking	Family tracking	Track location of family members	Family Tracker	Mutual location sharing
	Couple tracking	Consensual sharing of location and more	Couple Tracker	Syncs location, media content, SMS and call logs
	Friends tracking	Track friends if they are in vicinity	Friends Tracker	Like family tracker, and alerts if friend in vicinity
Subordinate tracking	Employee tracking	Track employees whereabouts	Where's my Staff	Similar to anti-theft
	Parental control	For parents to monitor their children	MMGuardian	Capabilities very similar to phone control
	Overt spyware	Claims to be spying app	Cerberus, mSpy, HelloSpy	Surreptitious phone monitoring & control

Fig. 5: Different categories of IPS-relevant apps and their typical capabilities.

sources" and disable Google Play Protect [4] regular scans. The link to download the app's APK is then found via browser or sent in an SMS link. As mentioned in Section III, there are many resources online that provide step-by-step instructions on how to do this. Installation and configuration usually takes only a few minutes of access to the victim's phone.

Remote installation of dual-use apps is possible from the Google Play web interface if the abuser knows the credentials of the device's primary Google account. However, Android enforces that no third party apps — those not packaged with the OS — can run until they are first opened on the device. The abuser must also grant permissions (for GPS, SMS and call logs, camera, and microphone, etc.) to either on or off-store apps, otherwise Android will not allow the app to access this information. Thus, for all apps we analyzed, an abuser needs to have physical access to the device at least once to perform activation. The exception here is when a dual-use app comes packaged with the OS, such as a family tracker provided by a smartphone manufacturer or cellular providers. We discuss these special cases in Section IV-C.

Once the app is installed and the permissions are granted, the abuser links the victim device to their credentials so they can access it remotely. Credentials may be a username and password, or a license number (for apps that require a paid subscription). All of the off-store spyware we analyzed can be configured to hide the app icon from the app drawer. Two of the 61 on-store apps we analyzed had this feature as well (Cerberus and TrackView).

Depending on the type of IPS app, the abuser is able to access gathered data in different ways. Most personal-use apps simply forward data to an email or a phone number that the abuser controls. Mutual trackers generally require installation of the app on two phones, one used by the victim and one used by the abuser. Some subordinate tracking apps also require a complementary app, but the majority offer web portals for accessing information from the target device. We discovered that several portals have simple but severe vulnerabilities that allow an arbitrary user of the spyware service to access sensitive information taken from *any* victim phone, and not just the ones associated with the abuser's account. We repeatedly attempted to disclose these vulnerabilities to the vendors, but never received a response.

No app that we analyzed required rooting the victim's phone [7], which is a technically sophisticated process for average users and is difficult using only software for Android 6.0 or above. That said, many off-store spyware apps offer additional functionality should the device be rooted, most notably the ability to read contents of messaging apps such as WhatsApp (which can't be done without root access). Some companies (e.g., FlexiSpy) sell phones that have their software pre-installed (with customized versions of Android or phones already rooted or jailbroken), providing a streamlined abuser experience with the most invasive monitoring abilities. As abusers often purchase and pay for the phone used by survivors [29], this is an acute threat. In summary, installation and use of IPS apps is easy for abusers, and gives them dangerous surveillance capabilities.

B. App Capabilities

Both on-store and off-store apps provide a shocking array of capabilities ranging from simple location tracking to near-complete remote control over a phone. We separate our discussion into three dimensions: monitoring abilities (what information is being extracted), covertness, and control.

Monitoring abilities. Most fundamental to IPS is an app's ability to monitor a victim's device. IPS apps typically gather a subset of the following types of information: location, communication logs (SMS and call logs), communication data (SMS content or call recordings), media content (photos, videos, or other files stored on the device), and phone usage (app usage or web history). In addition to passively gathering information, many apps can take photos or record ambient sounds in real time in response to an abuser's remote command.

Most basic dual-use apps are GPS tracking apps that record the location of the device and sync it with a remote server. A user can log into the remote portal to locate the device. Some dual-use apps, such as family locator apps, allow sharing this location data, and therefore enable mutual tracking among family members or friends. Most versions of Android and iOS ship with a built-in find-my-phone functionality; we discuss these apps in Section IV-C. Many third party find-my-phone apps, such as Find My Android, dispense with a remote server;

instead they are triggered by an SMS with a code-word and respond via SMS with the device's location.

Anti-theft apps, built to recover stolen phones, provide functionality beyond location tracking. For example, Wheres My Droid can take photos, record ambient audio, and wipe or lock the device remotely in stealth mode. It sends a notification if the SIM card is changed, sends full call and SMS logs, and sends GPS location of the phone if it is low on battery. Cerberus, another app built for anti-theft, provides all that functionality, along with a remote Android shell in its web portal. This means almost anything that can be done while using the phone can be done remotely. As previously discussed, Cerberus is recommended for IPS in blogs and forums. Similarly, survivors and professionals working with them have indicated that anti-theft apps are used in IPS [23, 29, 53].

Basic data syncing apps synchronize information across devices. A common personal use example is SMS forwarding with apps such as Mysms, which in an IPS context allows an abuser to monitor text messages. There are other file synchronization apps that will automatically copy one or more configurable folders (set during installation) to a cloud location. While these may seem benign, at least one IPS-related forum we found suggests using such an app in conjunction with a call recording app (that automatically records all incoming and outgoing calls) to listen in on a victim's communications.

Couple tracking apps are designed for mutual tracking, and tend to provide both location and data syncing. For example, Couple Tracker, which must be configured on a pair of phones, automatically shares with the other party location history, call logs, the first 30 characters of every sent and received SMS, and even Facebook activity (if provided with the credentials).

Phone control and child monitoring apps often provide some of the richest capabilities. Phone control apps are built for a user to remotely control their own phone for convenience, while parental control apps are meant for parents to keep an eye on their child's phone activity. Both types of apps provide access to location, SMS contents, call logs (sometimes recordings), all media contents, app usage, Internet activity logs, and even keylogging. Some apps can be configured to send notifications when the monitored phone engages in certain activities, like leaving a set geofence or calling a specific number. We note that all of the off-store spyware apps that we analyzed describe child safety as one of their use cases. An off-store app called TeenSafe (not found in our abuser-oriented searches), makes it difficult to use for IPS by checking the age of the Google account to which the device is registered. Abusers complain in reviews of TeenSafe about the difficulty in using it for IPS.

Covertness. In an IPS context, it's beneficial to an abuser if tools are covert, meaning they can operate without the victim's knowledge, and can potentially remain undiscovered even if the victim looks through all the apps in their app menu. Here we examine how difficult it would be for a victim, assumed to be of average technical sophistication, to notice the IPS app. In Section VI we discuss software tools for detecting spyware.

The Google Play developer policy obligates apps to "Present users with a persistent notification and unique icon that clearly identifies the app" [1] whenever the app is collecting sensitive information. Certain notifications are enforced by the operating system, such as the GPS usage notification icon that appears in the dock at the top of the screen whenever an app is using location service. This icon does not specify which app is using GPS, and is ever-present for many Android users. Other notifications are not OS-required, for example we encountered apps that by default do not display any notification when using the camera or microphone.

Even when notifications are present, we suspect victims are unlikely to observe them, let alone properly interpret their meaning. Prior work has shown how poorly users respond to other types of security indicators (e.g., the TLS lock in browsers [13, 52]).

Almost all off-store apps and even some on-store apps can be configured to hide their icons. (The OS does not enforce that an icon be displayed.) One off-store example is iKeyMonitor, which allows icon hiding, and can be later accessed by dialing #8888$*$ (an abuser can set the secret). An on-store app called TrackView leaves no access point on the device once the icon is hidden, but allows all of the app's settings to be changed from an app on the abuser's phone. Cerberus is another on-store app that hides its icon.

Control. Some spyware apps allow an abuser to remotely control the device. Child safety apps can be configured to block specific apps, impose browser restrictions, or limit the number of hours the phone can be used in a day. Anti-theft apps allow remotely locking the phone or wiping all data from the phone. Some apps, broadly classified as phone control apps, allow the abuser to remotely change the phone's settings, such as (re-)enabling GPS or WiFi.

Apps that allow such control of a device rely on commands being sent either through the customer's web portal (and thereby the company's server, which then relays the command to the device) or by sending an SMS to the phone containing a keyword that triggers a response from the app (the spyware passively observes all incoming SMS). Most of these apps allow the customer to customize their SMS keywords and may even hide the SMS from view in the UI.

C. Bundled dual-use apps

An important class of dual-use apps that fall outside the dichotomy of on- or off-store apps are the tools packaged with the OS, either by a cellphone manufacturer or a cellular service provider. One example of the latter is the Verizon Family Locator. These do not require an abuser to install an app on the phone, and often can be remotely activated with the credentials attached to the account that pays the cellular bill. Android natively provides tracking functionality, via Find My Device, or via Google Maps' unlimited location sharing functionality. Assuming the abuser has access to the victim's Google credentials, the abuser can remotely turn on the Google

Maps Timeline feature and obtain periodic (even historical) information about the victim's location. Google Drive and iCloud provide data syncing functionality to the cloud, and could be abused for extracting data from the device.

Some bundled apps that we investigated show notifications to the current user of the device. For example, Find My Device sends a notification stating that your device is being tracked. Adding a member (in an abuse context, the victim) in the Sprint Family Locator will send an activation SMS to the victim's phone. Even in these cases, as mentioned, notifications can be ignored or suppressed should the attacker have temporary physical access to the device.

These apps can be impossible to uninstall as they are bundled with the OS; at best they can be disabled. Looking ahead to mitigation, these apps will require different approaches than that used for on-store or off-store apps. See the discussion in Section VII.

V. Evidence of Developers' Complicity

In this section, we investigate the use of dual-use apps for IPS. The makers of some of these apps are not only aware of such abuse but are actively supporting the IPS use case via advertisement, by failing to refuse a potential customer that wants to use their software illegally, or failing to help an ostensible IPS victim being monitored by their software.

A. User Comments

On Google Play users can leave reviews of apps they have downloaded. We collected 464,625 reviews from over 9,000 apps. We searched for reviews mentioning both an intimate partner (husband, wife, boyfriend, bf, gf) and an IPS action word (track, spy, cheat, catch), and manually analyzed the results. We found 103 reviews on 82 apps that explicitly mention that the app is used for tracking or spying on a current or previous intimate partner. For example, a comment left for SMS Tracker Plus, an app which claims to be for parental control, states: *"Love it!!! I've been suspecting my gf cheating and this gave me answers really quick kick the curb girl"*. Another comment on ATTI Shadow Tracker, an app which markets itself for tracking a fleet of long-haul truckers, states: *"Love it! I can now keep an eye on my possibly cheating wife!"*. While we cannot verify the content of these reviews, we have no reason to suspect that they are dishonest.

B. Advertising

We found that many IPS apps, including dual-use ones, advertise IPS use cases directly or indirectly.

Google search advertisements. We searched Google with a subset of 1,400 queries from the 10,000 terms we found in Section III-A and recorded the ads shown on the first page of the search results. We found thousands of ads shown for search terms that show explicit intention of IPS, e.g., "how to catch a cheating spouse with his cell phone". A detailed analysis of advertisements shown on Google searches is given in Appendix D.

The ad texts often indicates that companies are advertising IPS as a use case. An ad recorded on March 10th 2017 for mSpy says *"Catch Cheater with mSpy App for Phones. Invisible Mode. Track SMS Chats Calls Location & Other. 1.000.000+ Satisfied Users. Try Now!"* Another ad recorded the same day for FoneMonitor reads *"Track My Wife's Phone — Want to Spy on your Wife? Track your Wife without her knowing. Discover Who Are They messaging. Download! 24-Hour Support Price Superiority No Jailbreaking and App Results Guaranteed."*

We informed Google about the IPS search terms that showed ads during our experiment. In response, Google expanded their restriction of ad serving on those types of search terms. We confirmed that ads are not being shown on explicit IPS search terms at the time of the final version of the paper.

Play Store. The Google Play website does not serve advertisements, but the Play Store app does. We chose some of the malicious terms from the snowball set and did manual searches in the Play Store app on an Android device. We found that apps on Play Store were also advertising on search terms like "phone spy on husband" or "see who bf is texting without him knowing." While a more systematic study is needed, it is clear that apps are being (and are allowed to be) advertised for IPS-related searches. After we shared the result of our study, Play Store has also expanded their restriction of ad serving on those types of search terms.

Blogs as IPS advertising. As mentioned in Section III-A, we found that Google searches such as "how can I read my wife's texts" yielded many blogs and forums providing advice. Some of these "blogs" were hosted on the domain of a dual-use app and explicitly outline why their product is ideal for covert tracking, sometimes accompanied by imagery of a battered woman and verbiage such as *"men need to have control of their families"*. An egregious example appears in Figure 1. These pages then link back to the site of the app, which is hosted at the same domain but in some cases have a completely different page format.

In addition to such advertising sites that appear on the same domain as an app's, we identified many ostensibly unaffiliated websites, blogs, and forums that serve the sole purpose of directing those interested in conducting IPS to a specific dual-use app. As one example, catchthemtoday.com, a blog focused on IPS has content such as *"Don't Be A Sucker Track Your Girlfriend's iPhone Now: Get It Here: Catch Her Today"*. The last three words are a Bitly link to appitunes.blogspot.com, which automatically redirects to mSpy.com. The blogspot.com page claims to have been last updated in October, 2017. It includes text that mSpy was created for child safety and employee monitoring uses, and that *"mSpy is not liable for other ways of the software use"*. (The redirect, however, ensures that this disclaimer is never seen.) The mSpy webpage, though, only discusses the child safety and employee monitoring use cases.

Direct marketing. During our manual analysis of apps in Section IV, we created accounts with, but had not yet purchased subscriptions for, several of the companies whose

customer service departments we later engaged (see below). We found that our dummy account received marketing emails encouraging us to sign up for their services once our free trials had expired. Two companies encouraged us to buy a smartphone from them which would come prepackaged with their software (for streamlined abuser experience). We saw comparable advertisements for iOS. While these emails don't explicitly suggest giving these devices to an intimate partner, these same companies do advertise those use cases elsewhere.

C. Customer Support

Though some companies actively advertise IPS as a use case, we suspected that many more turn a blind eye to it. To investigate this, we contacted customer service departments posing both as an abuser and a survivor to see how they responded. Our IRB confirmed that this is exempted from review, as the interaction is in their professional capacities and does not collect any PII. We selected a subset of 12 apps, 5 off-store and 7 on-store. We chose a set of apps that had shown advertisements on IPS-related terms, or that we knew to be widely used, either by install counts for on-store apps or by blog and forum mentions for off-store apps.

To investigate the attitude of a company towards the use of their software for IPS, we initiated live chats or email conversations with their customer support teams, posing as a potential abuser. Only 11 of the 12 apps offered any way to contact them.

All conversations were initiated from a Gmail account associated with the name Jessica and began with the following prompt:

> *Hi, If I use this app to track my husband will he know that I am tracking him? Thanks, Jessie*

If customer service responded to this prompt with a question, such as "What kind of phone does he have?" we continued the conversation in an attempt to elicit the company's attitude towards this use case.

Of the 11 companies we contacted, 2 have yet to respond. Of the 9 that responded, one, TeenSafe, which is an off-store app, delivered a strong admonishment and legal warning about IPS. The other 8 responded with some version of "No, he shouldn't be able to tell", making them complicit in potential abuse. The customer support representative for TrackView, which is available on Play Store and has an entire Google Groups Forum for customer support, told us that with the paid plan, the icon and all notifications could be hidden. We have confirmed that this is true. TrackView is also the only company that has responded to our inquiries posing as a survivor looking for help removing their app. Their response showed no sign of concern, and their advice of "look in the app menu and delete it" was not useful, given that the app icon was concealed.

VI. Inefficacy of Existing Anti-Spyware

The previous sections reveal the prevalence, ease-of-use, and severity of overt spyware and dual-use apps. Moreover, many of the tools are inherently, or can be configured to be, difficult to detect by inspection of the device via the normal UI. What can potential victims of spyware do? Current best practice is circumstantial [30], with victims advised to suspect spyware should there be spikes in bandwidth usage, decreased battery life, slow responsiveness, or information the abuser knows that is seemingly only possible to learn from spyware. Typically, the only recourse for strong suspicions are factory resetting or completely discarding the phone. Obviously it would be better to have technical means for detecting and mitigating spyware.

A number of tools do advertise the ability to detect and remove spyware, perhaps suggesting defenses against spyware are close at hand. These anti-spyware tools range from mobile versions of well-known, commercial anti-virus systems such as Avast, Norton, and ESET, down to barely functional apps that appear to be scams. In this section we put these countermeasures to the test to see whether they should be used by potential victims.

A. Anti-Spyware Tools on Google Play

There are many apps in the Google Play store that claim to be anti-spyware tools. To identify these apps we followed a similar procedure to that used for discovering spyware, but this time performing searches from a potential spyware victim's perspective. We began our query snowball with the terms "anti spyware", "remove phone tracker", and "spyware removal tool", and conducted snowball querying using the query completion API provided by Google Play (see Section III-B). The eventual snowball size was 13, and upon search with those terms, returned 147 apps that have more than 50,000 installations as reported by Google Play. Manual inspection of the 147 apps revealed 40 to be relevant for removing spyware. All of them advertise a free scanning facility, but some charge money to remove apps.

Among these 40 apps 7 were from major antivirus vendors: Avast, AVG, Avira, ESET, Kaspersky, McAfee, and Norton. The remaining 33 apps are from other vendors, though note that some of these have more than 100 million downloads. In Figure 6 we show the 19 anti-spyware apps that were downloaded at least 10 million times or came up in the top 10 results for searching "anti spyware" in Play Store, as recorded in November 2017.

Interestingly many anti-virus apps provide find-my-phone, anti-theft, or family safety functionality, making these potentially dual-use. None are covert, but even so these anti-spyware tools could hypothetically be used by abusers as dual-use apps. Nevertheless we do not consider them as such, because their primary functionality is not for spying (see Section II). More pragmatically, they are not returned in response to abuser search queries and we found no evidence online or in prior work of their abuse in IPS settings.

Experimental setup. To evaluate the efficacy of the anti-spyware apps in detecting dual-use apps, we installed 276 dual-use apps out of 280 identified via manual inspection as described in Section III-B on a device running Android 6.0 (Marshmallow). Four could not be installed due to compatibility issues. We also installed 20 out of the 23 off-store spyware

Anti-spyware tool	D/L (mn)	On-store (276)	Off-store (20)	Benign (100)
360 Security	100	2	80	0
Anti-virus Dr.Web	100	2	70	0
Avast Mobile Security[1]	100	2	70	0
AVG Antivirus[1,2]	100	2	70	0
DFNDR Security	100	2	85	0
Lookout Security	100	3	75	0
ALYac	10	2	70	0
Antivirus (TrustGo)	10	2	80	0
Antivirus (TrustLook)	10	2	70	0
Avira[1]	10	3	60	0
Kaspersky[1]	10	1	85	0
Malwarebytes[2]	10	3	85	0
McAfee Mobile[1,2]	10	2	**90**	0
ESET[1]	10	1	14	0
Norton Mobile[1,2]	10	**13**	70	2
Virus Cleaner[2]	10	2	75	0
Anti Spy Mobile[2]	1	**47**	**95**	**12**
Incognito[2]	1	2	5	0
Anti Spy (skibapps)[2]	< 1	**36**	73	**10**
Others (average over 21 apps)	1	2	70	0
Virustotal (3+ AVs)	N/A	7	100	3

[1] Apps from popular antivirus providers.
[2] Apps among top 10 search results in Play Store for "anti spyware".

Fig. 6: True positive (third and fourth columns, higher is better) and false positive (final column, lower is better) detection rates (in percentages) of anti-spyware apps available in the Play Store ordered by reported number of downloads (second column). The final row reports on using Virustotal to flag an app if at least three AV engines flag the app.

apps we collected outside the Google Play store. Again the remaining three could not be installed due to compatibility issues. Finally to measure false positive rates, we installed the 100 top-selling apps (in November 2017) from Google Play that are not usable as spyware (manually verified).

For each anti-spyware app, we first install the app, allow it to complete its scan of the device, record its results, and then uninstall the anti-spyware app. The output format from the anti-spyware apps varies and often does not provide a report that can be exported programmatically, so we manually transcribe the results. Some anti-spyware apps give binary classification, while others provide multiple types of classifications. For example, Norton Anti-Virus categorizes apps as "ok", "malware", "medium privacy risk", and "high privacy risk". Whenever an app offered classification more granular than binary, we counted anything not marked "ok" as being flagged as spyware.

Evaluation. Of the 40 anti-spyware apps, 37 are completely ineffective against dual-use apps, flagging at most 3% of them. Most of the anti-spyware apps flag more than 70% of the off-store spyware apps. The performance results of the anti-spyware apps are given in Figure 6.

The ones that detect the most spyware have higher false positive rates. For example, Anti Spy Mobile catches more than 47% of on-store IPS-relevant apps, but flags Chrome, Play Store, and Amazon apps as risky. Further investigation revealed that these anti-spyware apps simply mark any app that uses certain permissions as risky.

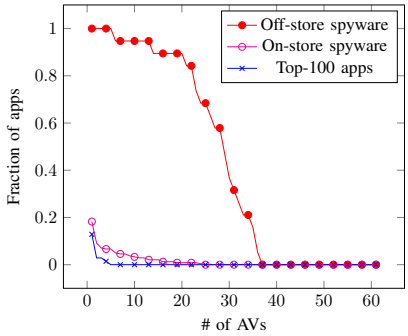


Fig. 7: Fraction of off-store spyware, on-store spyware, and top-100 apps (benign, non-spyware apps) detected by the indicated number of AV engines available in Virustotal.

All but one of the top-brand anti-virus providers (e.g., Avast, AVG, Avira, ESET, McAfee, and Kaspersky) detect less than 3% of dual-use apps. Presumably this reflects their design goals, which do not necessarily include detecting IPS spyware, let alone dual-use apps. Indeed in non-IPV contexts marking some of the dual-use apps as malicious would represent false positives to their users.

B. *Virustotal Analysis*

We also evaluate whether Virustotal [57], an aggregator of many anti-virus engines, can be used to identify IPS apps. Virustotal hosts more than 60 anti-virus engines (AV engines), and a large number of tools for static and dynamic analysis of content. Access to the Virustotal API is free for non-commercial use and takes as input an MD5 or SHA1 hash of an app's binary.

We had Virustotal evaluate the 280 on-store apps identified in Section III-B as well as the 23 off-store apps. Figure 7 gives the fraction of on-store and off-store IPS apps and the top 100 benign apps flagged by at least the indicated number of AV engines. As one datapoint, only 21 apps out of 280 on-store dual-use apps (8%) were flagged by at least three AV engines. The best two AV engines were Cyren and WhiteArmor. Cyren flagged 6% of the on-store IPS apps, and 70% of the off-store spyware, but Cyren also flagged one of the top 100 apps (Pandora Radio). WhiteArmor flagged less dual-use apps than Cyren (only 5%), but flagged all of the off-store spyware, and did not have any false positives. In the end, we conclude that these engines are not designed to catch dual-use apps, their focus instead being on other forms of malware.

VII. Discussion: Dealing with IPS Spyware

Spyware used in IPV settings endangers the safety and privacy of survivors. As our measurements suggest, spyware takes on many forms, ranging from apps overtly designed and advertised for IPS to dual-use tools with functionalities easily repurposed for IPS. Existing anti-spyware tools do not sufficiently detect dual-use apps.

Thus, we propose a multi-pronged strategy for combating IPS apps and improving safety for survivors. A full investiga-

tion of our suggestions will require significant future work — here we outline ideas and discuss looming hurdles.

Improved detection and removal. An urgently needed first step is to improve detection of IPS apps. We envision building upon our measurement framework from Section III-B to construct a proof-of-concept blacklisting tool capable of detecting potential IPS apps. Apps need to be collected and labeled on an ongoing basis. An issue to be dealt with is ensuring the robustness of our crawling and detection infrastructure in the face of malicious developers seeking to avoid classification as spyware. For example, our ML classifier may be vulnerable to evasion attacks [36,41,42]. We hope that anti-virus vendors will extend their commercial tools to deal with IPS spyware and dual-use apps, perhaps using our techniques.

The deployment of detection and removal tools faces particular challenges in the IPV context, as using anti-spyware or removing IPS apps may risk escalation from digital abuse to physical violence (see [29,30,46]). This means deployment may require multiple modalities, such as a covert or easily-removed anti-spyware app, or even use of a USB-connected laptop at shelters (or other places that victims may go to obtain help) to scan the device. New guidelines for safety planning when IPS apps are found will be needed, as removing them may be too dangerous in the short term.

OS notifications and protections. We found many IPS apps, including on-store apps, work in the background without proper notification — even when sensitive data (i.e., camera, microphone, chat messages, photos) is being relayed. Two IPS apps also hide their icon in the app drawer. Though Google Play's developer policy explicitly prohibits this [1], there is no OS-level enforcement.

We propose that mobile OS developers strengthen user protections by enforcing the policies in their developer agreements. Bundled dual-use apps, such as Google Maps or iCloud should take special steps to regularly inform the user if they are syncing any sensitive data with a remote server. Future work would be needed to evaluate the efficacy of specific notification policies, as users do not always notice indicators like "recording lights" [27, 51]. OS-level protections would need to be carefully designed, so as not to be bypassable, at least should the device not be rooted (c.f., [15]). Finally, and most critically, these mechanisms would need to be carefully constructed to balance the needs of legitimate applications of dual-use apps with the threat of their use as spyware.

IPS use case prevention. Our measurements revealed that dual-use apps are often advertised for IPS, both in paid advertising channels and in organic marketing (e.g., blog posts and search results). Some developers explicitly condone IPS, while in other cases they do little to prevent it. We believe ad networks, OS vendors, and developers can work together to better prevent use of legitimate dual-use apps for IPS.

As a start, advertising networks should stop accepting paid ads for search terms related to IPS/IPV. There is precedence for this in other contexts, such as prescription drug advertising [55]. In response to this paper, Google has already stopped showing advertisements on IPS-related search terms. Search engines could also potentially preference information about legality in response to abuse-related queries, hopefully creating a deterrent for an abuser. Organic marketing will be difficult to curb from a technology perspective, but here law enforcement agencies, such as the FTC and DOJ in the United States, might escalate their enforcement of policies against products intended for illegal use, such as in the CyberSpy case [20,47].

Future work could develop guidelines for building apps which are less attractive for IPS use cases. As an example, parental control apps need not be surreptitious, so they could be made conspicuous. Similarly, SMS syncing apps should always show clear notifications when forwarding to another device, as done by some messenger apps already [31]. Finally, developers could better monitor comments and reviews, and refuse to continue service to people indicating an IPS usage. For example, TeenSafe's customer service refused to help us when we indicated we intended to use their app for IPS.

VIII. Conclusion

In this paper we provided the first in-depth measurement study of the ecosystem of software used for IPS on mobile devices. Taking the view of an abuser, we used manual as well as automated crawling to document the abundant resources currently available to abusers, including how-to guides, question-and-answer forums, and apps. Over a one month period, we crawled the Google Play Store and used a combination of manual review and machine learning to discover a large amount of dual-use apps: those designed for some legitimate use, but which are being repurposed by abusers for IPS. By investigating advertising behavior, online forum discussions, and customer service responses, we showed that many dual-use app vendors are tacitly facilitating or, in some cases, condoning IPS. We measured the efficacy of existing anti-spyware tools and found them insufficient for use in IPV contexts.

Given that abusers use IPS apps to cause emotional and physical harm, including even murder, there is an acute need for the security community to help mitigate the threat. In response to our paper, Google improved safety for their users by taking action against apps that violated Play Store policies. They have also increased restrictions on advertisement serving for IPV-related queries. More broadly, we have initiated discussion about future work ranging from improved detection tools to legal and regulatory improvements. We hope future research and advocacy will further increase digital security and safety for those suffering in IPV situations.

IX. Acknowledgments

We thank Kurt Thomas and others at Google for their feedback, as well as the anonymous reviewers for their insightful comments. This work was supported in part by NSF grants 1619620, 1717062, 1330308, 1253870, and 1514163, as well as gifts from Comcast, Google, and Microsoft.

REFERENCES

[1] "Developer policy center - privacy and security - malicious behavior," https://play.google.com/about/privacy-security/malicious-behavior/, accessed: 2017-09-13.

[2] "FlexiSpy," https://www.flexispy.com, accessed: 2018-02-12.

[3] "Google Play - Apps," https://play.google.com/store/apps/, accessed: 2018-02-13.

[4] "Google Play Protect," https://www.android.com/play-protect/, accessed: 2018-02-13.

[5] "lxml - XML and HTML with Python," http://lxml.de/, accessed: 2018-02-12.

[6] "mSpy," https://www.mspy.com, accessed: 2018-02-12.

[7] "Rooting (Android)," https://en.wikipedia.org/wiki/Rooting_(Android), accessed: 2017-11-12.

[8] "Hearing before the subcommittee on privacy, technology, and the law, Senate, 113th Cong. (testimony of Brian Hill)," p. 27, 2014.

[9] "Spyware use in domestic violence 'escalating'," http://www.bbc.com/news/technology-30579307, 2014.

[10] "Spyware's role in domestic violence," http://www.smh.com.au/technology/technology-news/spywares-role-in-domestic-violence-20140321-358sj.html, 2014.

[11] "Alternativeto - crowdsourced software recommendations," *Online: https://alternativeto.net/*, 2017.

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

[13] D. Akhawe and A. P. Felt, "Alice in Warningland: A large-scale field study of browser security warning effectiveness." in *USENIX security symposium*, vol. 13, 2013.

[14] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket." in *Ndss*, vol. 14, 2014, pp. 23–26.

[15] M. Brocker and S. Checkoway, "iSeeYou: Disabling the MacBook webcam indicator LED," in *USENIX Security Symposium*, 2014, pp. 337–352.

[16] S. C. Burke, M. Wallen, K. Vail-Smith, and D. Knox, "Using technology to control intimate partners: An exploratory study of college undergraduates," *Computers in Human Behavior*, vol. 27, no. 3, pp. 1162–1167, 2011.

[17] R. V. Chandra and B. S. Varanasi, *Python requests essentials*. Packt Publishing Ltd, 2015.

[18] D. K. Citron, "Spying Inc." *Wash. & Lee L. Rev.*, vol. 72, p. 1243, 2015.

[19] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[20] U. F. T. Commission, "Spyware seller settles FTC charges; Order bars marketing of keylogger software for illegal uses," https://www.ftc.gov/news-events/press-releases/2010/06/spyware-seller-settles-ftc-charges-order-bars-marketing-keylogger.

[21] J. Cox, "Google pushed illegal phone spyware to snoop on your spouse," 2017, https://www.thedailybeast.com/google-pushed-illegal-phone-spyware-to-snoop-on-your-spouse.

[22] ——, "Meet FlexiSpy, the company getting rich selling 'stalkerware' to jealous lovers," https://motherboard.vice.com/en_us/article/aemeae/meet-flexispy-the-company-getting-rich-selling-stalkerware-to-jealous-lovers, 2017.

[23] J. P. Dimond, C. Fiesler, and A. S. Bruckman, "Domestic violence and information communication technologies," *Interacting with Computers*, vol. 23, no. 5, pp. 413–421, 2011.

[24] W. Enck, P. Gilbert, S. Han, V. Tendulkar, 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," *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, p. 5, 2014.

[25] B. Farinholt, M. Rezaeirad, P. Pearce, H. Dharmdasani, H. Yin, S. Le Blond, D. McCoy, and K. Levchenko, "To catch a ratter: Monitoring the behavior of amateur darkcomet rat operators in the wild," in *Security and Privacy (SP), 2017 IEEE Symposium on*. Ieee, 2017, pp. 770–787.

[26] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, ser. SPSM '11. New York, NY, USA: ACM, 2011, pp. 3–14. [Online]. Available: http://doi.acm.org/10.1145/2046614.2046618

[27] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, ser. SOUPS '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:14. [Online]. Available: http://doi.acm.org/10.1145/2335356.2335360

[28] C. Fraser, E. Olsen, K. Lee, C. Southworth, and S. Tucker, "The new age of stalking: Technological implications for stalking," *Juvenile and family court journal*, vol. 61, no. 4, pp. 39–55, 2010.

[29] D. Freed, J. Palmer, D. Minchala, K. Levy, T. Ristenpart, and N. Dell, "Digital technologies and intimate partner violence: A qualitative analysis with multiple stakeholders," *PACM: Human-Computer Interaction: Computer-Supported Cooperative Work and Social Computing (CSCW)*, vol. Vol. 1, no. No. 2, p. Article 46, 2017.

[30] ——, ""a stalker's paradise": How intimate partner abusers exploit technology," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018.

[31] K. Gander, "WhatsApp Web: Messaging client now available on Internet browsers," *The Independent*, 2015.

[32] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "RiskRanker: scalable and accurate zero-day android malware detection," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 281–294.

[33] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[34] M. Hatada and T. Mori, "Detecting and classifying Android PUAs by similarity of DNS queries," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, July 2017, pp. 590–595.

[35] Hello Spy, "Mobile Spy App for Personal Catch Cheating Spouses," 2017, https://web.archive.org/web/20180305193744/http://hellospy.com/hellospy-for-personal-catch-cheating-spouses.aspx?lang=en-US.

[36] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[37] J. Koebler, "'I see you': A domestic violence survivor talks about being surveilled by her ex," https://motherboard.vice.com/en_us/article/bmbpvv/i-see-you-a-domestic-violence-survivor-talks-about-being-surveilled-by-her-ex, 2014.

[38] P. Kotzias, L. Bilge, and J. Caballero, "Measuring PUP prevalence and PUP distribution through pay-per-install services." in *USENIX Security Symposium*, 2016, pp. 739–756.

[39] S. Le Blond, A. Uritesc, C. Gilbert, Z. L. Chua, P. Saxena, and E. Kirda, "A look at targeted attacks through the lense of an ngo." in *USENIX Security Symposium*, 2014, pp. 543–558.

[40] K. E. Levy, "Intimate surveillance," *Idaho L. Rev.*, vol. 51, p. 679, 2014.

[41] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 641–647.

[42] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," in *CEAS*, 2005.

[43] W. R. Marczak, J. Scott-Railton, M. Marquis-Boire, and V. Paxson, "When governments hack opponents: A look at actors and technology." in *USENIX Security Symposium*, 2014, pp. 511–525.

[44] E. Mariconti, L. Onwuzurike, P. Andriotis, E. De Cristofaro, G. Ross, and G. Stringhini, "MaMaDroid: Detecting Android malware by building Markov chains of behavioral models," *arXiv preprint arXiv:1612.04433*, 2016.

[45] J. Martin and N. Christin, "Ethics in cryptomarket research," *International Journal of Drug Policy*, vol. 25, pp. 84–91, 2016.

[46] T. Matthews, K. O'Leary, A. Turner, M. Sleeper, J. P. Woelfer, M. Shelton, C. Manthorne, E. F. Churchill, and S. Consolvo, "Stories from survivors: Privacy & security practices when coping with intimate partner abuse," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 2189–2201.

[47] U. D. of Justice, "Man pleads guilty for selling "StealthGenie" spyware app and ordered to pay $500,000 fine," https://www.justice.gov/opa/pr/man-pleads-guilty-selling-stealthgenie-spyware-app-and-ordered-pay-500000-fine, 2014.

[48] F. Olano, "Google-play-scraper," https://github.com/facundoolano/google-play-scraper, 2017.

[49] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie, "WHYPER: Towards automating risk assessment of mobile applications." in *USENIX Security Symposium*, vol. 2013, 2013.
[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
[51] R. S. Portnoff, L. N. Lee, S. Egelman, P. Mishra, D. Leung, and D. Wagner, "Somebody's watching me?: Assessing the effectiveness of webcam indicator lights," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 1649–1658.
[52] R. Shah and K. Patil, "Evaluating effectiveness of mobile browser security warnings," *ICTACT Journal on Communication Technology*, vol. 7, no. 3, pp. 1373–1378, 2016.
[53] A. Shahani, "Smartphones are used to stalk, control domestic abuse victims [web log post]," 2014.
[54] S. G. Smith, K. C. Basile, L. K. Gilbert, M. T. Merrick, N. Patel, M. Walling, and A. Jain, "The national intimate partner and sexual violence survey (NISVS): 2010-2012 state report," 2017.
[55] U. States Department of Justice, "Google forfeits $500 million generated by online ads & prescription drug sales by canadian online pharmacies," https://www.justice.gov/opa/pr/google-forfeits-500-million-generated-online-ads-prescription-drug-sales-canadian-online, 2011.
[56] K. Thomas, J. A. E. Crespo, R. Rasti, J. M. Picod, C. Phillips, M.-A. Decoste, C. Sharp, F. Tirelo, A. Tofigh, M.-A. Courteau *et al.*, "Investigating commercial pay-per-install and the distribution of unwanted software." in *USENIX Security Symposium*, 2016, pp. 721–739.
[57] V. Total, "VirusTotal: Free online virus, malware and URL scanner," *Online: https://www.virustotal.com/en*, 2012.
[58] D. Woodlock, "The abuse of technology in domestic violence and stalking," *Violence against women*, p. 1077801216646277, 2016.
[59] D.-J. Wu, C.-H. Mao, T.-E. Wei, H.-M. Lee, and K.-P. Wu, "Droidmat: Android malware detection through manifest and API calls tracing," in *Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on*. IEEE, 2012, pp. 62–69.
[60] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: android malware characterization and detection using deep learning," *Tsinghua Science and Technology*, vol. 21, no. 1, pp. 114–123, 2016.
[61] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on imbalanced data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 80–89, 2004.
[62] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh, *Taming Information-Stealing Smartphone Applications (on Android)*. Springer Berlin Heidelberg, 2011, pp. 93–107.

APPENDIX

A. Query snowballing, seed queries, and initial filtering

We use a query snowballing technique to find a comprehensive set of search terms using the recommendation APIs provided by search engines. In Figure 8, we give the pseudocode of our snowballing approach. The algorithm takes as input a set of seed queries and the maximum number of queries to return. We use different seed queries for Google and Google Play, because of the different nature of the query recommendation APIs. While the former suggests semantically related queries, the latter provides only query completion. The initial seed queries for both searches are listed in Figure 9.

Both the recommendation APIs return lots of unrelated search terms, more so in Google than in Google Play. For example, "cheating" is expanded to a game "cheating tom", or "cheating husband" is expanded to "cheating husband quotes", none of which is what we are looking for. As we expand the search results repeatedly, unrelated queries will drive the snowball away from related terms. We therefore filter the suggested queries that contain any of the words (regular expressions) listed in Figure 9 (right table).

QSnowball(Q_{init}, ℓ):
$Q \leftarrow \phi$
while $|Q_{\text{init}}| > 0$ do
  $q \leftarrow \mathsf{pop}(Q_{\text{init}})$; $Q \leftarrow Q \cup \{q\}$
  If $|Q| \geq \ell$ then return Q
  $Y \leftarrow \mathsf{recommend}(q)$
  $Y' \leftarrow \mathsf{filter}(Y) \setminus Q$
  $Q_{\text{init}} \leftarrow Q_{\text{init}} \cup Y'$
return Q

Fig. 8: Query snowballing approach. Here recommend represents a query recommendation API that takes as input a query string and returns a set of recommended queries and filter uses regular expression heuristics to remove unrelated search phrases. The set Q_{init} is an initial set of query phrases and ℓ is the limit of the size on the query snowball Q.

B. IPS apps in iTunes App Store

During our Google search in Section III, we found several dual-use apps enlisted on the Apple's app store. All the off-store apps that we found support both Android and iOS platforms. To seek further evidence of IPS apps, we decided to apply our measurement pipeline to the iOS platform using our crawling pipeline from Section III with only a few modifications, as we discuss bellow.

Interestingly, unlike Play Store API, the search API provided by iTunes App Store (official market place for Apple, also called App Store) does not return any search results or query suggestions on many IPS relevant search terms such as "catch your spouse cheating." However, Google's site-specific search (searching by adding a prefix: "site: itunes.apple.com" to the search term) returned several links to apps listed in the App Store that are not found via direct search in iTunes.[5] Also, we found iTunes query completion API returns a lot of completely unrelated queries not relevant to IPS. We therefore chose to use the snowball of search terms obtained using Google Play store search suggestions to search iTunes and Google for dual-use iOS apps.

Together via the direct search on iTunes and via site-specific search in Google, we found 2,724 apps. Manual investigation of descriptions of a random sample of 500 apps from this list revealed that 97 apps (20%) are capable of IPS. The fraction of on-store IPS apps we found in Apple is less than what we saw for Android (28%). Among the on-store dual-use apps enlisted in Google Play we found 9 IPS relevant apps listed in iTunes App Store too (with very similar app identifiers). However, many cross-listed apps have different functionalities in iOS due to its more restricted permission model and stricter enforcement of developer policy. For example, the TrackView app in iOS does not hide their app icon or use the camera surreptitiously, while in Android it does (as discussed in Section IV-B).

[5]We tried similar site-specific search to find Play Store apps too (by adding "site: paly.google.com"), but did not find any new app other than those we already found via direct search in Play Store.

Google Play	Google	Blacklist
cheater, cheating, cheating {agent}, catch cheating, catch cheating {agent}, catch {agent} cheating, track location, track {agent}, family tracker, {agent} tracker, track {agent} cheating, track {agent} phone, track phone, app for tracking location, find my phone, find my lost phone, find phone location, track my phone, call record, listen to call, gps tracking, read emails from {agent}'s phone, spy my {agent}, app for spying, apps for spying on my {agent}, spy on my {agent}, track sms, sms tracker, sync sms, read sms from another phone, message sync, phone control, control kids phone, track email, track messages, track calls, hidden keylogger, keylogger, keylogger for android, phone sync	how to catch my cheating {agent} app, how to track my cheating {agent} app, app for spying on my {agent}, app for tracking my {agent}, get location of another phone app, listen to calls from another device app, track my {agent}, track my {agent}s phone without them knowing, track location of my {agent} app, read sms from another phone app, app for tracking my kids, app for seeing my kids phone, keylogger for android, easy spy app for android, hidden spy app for android, app to see photos in my phone remotely, see all whatsapp messages app, see all facebook messages app, spyware for android devices, record calls app	game, sport, mile, gta, xbox, royale, golf, fit, food, flight, run, tracks$, car, cheating tom, cheat.*code, refund, cheatsheet, chart, cheat.*sheet, cheat.*engine, gas budd?y, calorie, money, expense, spending, tax, budget, period, diet, pregnancy, fertility, weight, gym, water, work ?out, track and field, exercise, cheats, baby.*photos, tv, time, hour, minute, day, month, year, sale, ski, sleep, walking, block, anti.*tracking, rent, nutrition, corporate, insta(gram)?, facebook, twitter, tinder, spyfall, forms?, exam, dhl, fedex, ups, read.*loud, quotes, ps4, ps3

Fig. 9: List of seed search terms (separated by ",") for conducting query snowballing with Google Play and Google search. Here {agent} is replaced with each of {boyfriend, girlfriend, wife, husband, spouse, partner}. On the **right** is the blacklist of words or regular expressions used to filter queries that have them.

App Store does not provide all the information we get from Play Store. Notably, in the App Store, permissions requested by an app are not available. Therefore we had to modify and retrain our machine learning algorithm separately for Apple. For training and cross-validation we used the 500 hand labeled apps mentioned above. The feature set was constructed using the app description, the app title, and the genres of that app as listed in the App Store. A bag-of-word model with pruning was constructed in the same way described in Section III-C. From the BoW model we pick the 1,100 most discriminatory features (1,000 from description and 100 from the title and the genres) based on χ^2-statistic [61].

In 10-fold cross validation using logistic regression (LR) model (with L_2 penalty, and inverse of regularization strength, C, set to 0.0385), we found the classifier can accurately classify 92% of apps, with a false positive rate of 7% and false negative rate of 8%. If we set the cutoff to 0.3 (as we did in case of Android), the false negative rate goes below 1% with 10% false positive.

C. Pruning with MTurk.

In Page 6 we show how we can tune machine learning to remove apps that are obviously irrelevant to IPS, leaving us with nearly 34% of all apps that ML classifier flags as "dual-use." However, among the apps flagged by the classifier, nearly 20% are falsely tagged (based on our hand-labeled training data). Therefore, we decide to use a second level pruning of false positive apps from the Google Play store leveraging Amazon's Mechanical Turk (MTurk) to rapidly employ a large pool of human workers to label apps as dual-use or not.

While our initial experiment does not produce results better than the ML classifier, it definitely testifies the possibility and opens up an interesting question of how to utilize a crowdsourcing framework to perform a non-trivial task such as identifying spyware apps based on their descriptions.

Pilot study to ensure feasibility. Though MTurk provides an efficient method for simple classification tasks, such as image tagging, our task is more nuanced, and could require domain knowledge from the workers to perform correctly. For example, the definition of a dual-use app is not always immediately apparent, and often relies on "what-if" judgments about potential app usage rather than any observable phenomena. In order to verify MTurk's viability for completing our classification task, we conducted a pilot study with a small set of workers.

		Ground Truth		
		dual-use	benign	**Total**
MTurk	dual-use	30	1	31
	benign	3	65	68
	Total	33	66	99

Fig. 10: Confusion matrix of MTurk labels (majority among 5 workers) and ground truth (researchers' labels) of 99 apps (randomly sampled from TR) from the pilot study.

As part of our required qualification test, we gave workers a short (i.e., a couple paragraphs) description of dual-use apps and examples of both benign and dual-use apps, including some "borderline" cases. We then asked workers to classify ten sample apps we hand-labeled beforehand as either benign or dual-use. We found that most workers (84.6%) were able to accurately classify all ten apps by their second attempt at the qualification test.

Once a worker passes the qualification test, the worker is allowed to accept actual classification tasks (HITs). Each task contains 3 apps (with a $0.06 reward for labeling each app) and must be completed by five different workers. For each app, we take the majority vote of the classification submitted from all five workers. To evaluate the promise of crowdsourced labels, we first performed a pilot study by submitting 99 randomly sampled apps from the TR set (data that we hand-labeled and used for training the ML classifier). We use Cohen's Kappa (κ) statistic [19] to compare the agreement between crowdsourced labels and the researcher-assigned labels.

In the pilot study, we found a promising agreement rate between the crowdsourced labels and the researchers' labels ($\kappa = 0.96$; the maximum possible value of κ is 1). This amounts to 95% of the labels matching across the 99 apps, and only a 1% false negative rate (taking the researchers' labels as the ground truth). In Figure 10 we note the confusion matrix of this experiment. The results suggest the viability of using crowdsourcing to identify dual-use apps.

Study with all the hand-labeled apps. Following the pilot

study, we submitted all of the remaining apps from our hand-labeled set of 1,200 apps (TR + TS_1 + TS_2) to MTurk. To expedite the data collection, we included 7 apps in each HIT for a total payment of 0.42 ($\$0.06 \times 7$) per assignment. All of the apps were labeled by five different workers within 48 hours. However, the final agreement rate was worse than the pilot study at $\kappa = 0.64$: only 85% of crowdsourced labels matched the researcher labels, with a 12% false negative rate.

We found that a small number of the workers mislabeled a relatively large number of apps. After removing all labels from workers with agreement rate $\kappa \leq 0.5$, we re-submitted the apps requiring more labels (using the same HIT format: 7 apps per HIT). We also modified our initial qualification test by giving more exemplary instruction of major classes of dual-use apps. After obtaining the new labeling the agreement of the MTurk majority with the ground truth improved to $\kappa = 0.76$. In Figure 11 we show the performance of the MTurk majority labeling.

Evaluation. To conjunct MTurk into our rest of the pipeline, we decided to use 0.3 as our classification threshold. With this threshold, we do not submit any negatively-labeled apps by our machine classifier to MTurk, and also the apps on which machine classifier's confidence is high (≥ 0.7). We only submit the positive-apps for which the classifier's confidence is low (≤ 0.7). For the rest of the apps, we will take the ML classifier's labeling as the final label.

The final performance of this pipeline is recorded in the Figure 11 (last row). Interestingly, the pipeline has consistently lower false negatives across all datasets than logistic regression with cutoff 0.5, while having similar false positive rate. Also, we found for test data TS_2, the accuracy is $> 97\%$, better than the best machine learning can achieve.

While the initial results are not very promising, we can improve on this. For example, given that the labeling dual-use apps require some domain knowledge, we can design a more nuanced worker-training process. Also, we can task workers to identify capabilities and purpose of the apps, instead of making a judgment call about whether or not the app is IPS relevant. For example, the worker finds out from the description whether the app can sync SMS, or can be used for parental control, etc. This information can be used to further classify those apps more accurately into IPS and benign categories. We leave a detailed analysis of this approach as future work.

D. Analysis of Google Ads on IPS search terms

We searched Google for ten days in October 2017 with a subset of 1,400 queries from the 10,000 terms we found in Section III-A. We searched from a Chrome browser on an OSX machine and recorded the contents of the first page of the search results. We did not set up any user profile, and performed each search from a new browser session (though persistent cookies were not purged). We extracted a total of 7,776 ad impressions associated with 214 domains. Among our search terms, 340 showed at least one ad during the measurement period. The term "how to catch a cheating spouse with his cell phone," served the most ad impressions associated to 18 different domains. The most common domain (truthfinder.com) appeared in 897 ad impressions across 112 different search terms.

		Training	Test (1st wk)	Test (4th wk)
	dual-use	280	28	22
	benign	720	72	78
Logistic Regression (cutoff: 0.5)	Accu.	96%	91%	95%
	FNR	4%	4%	10%
	FPR	4%	11%	6%
Logistic Regression (cutoff: 0.4)	Accu.	93%	88%	88%
	FNR	2%	4%	10%
	FPR	9%	15%	12%
Logistic Regression (cutoff: 0.3)	Accu.	86%	82%	81%
	FNR	$< 1\%$	0%	0%
	FPR	19%	25%	24%
MTurk (majority among 5)	Accu.	91%	89%	96%
	FNR	20%	11%	19%
	FPR	4%	11%	0%
Whole Pipeline	Accu.	96%	91%	97%
	FNR	4%	0%	5%
	FPR	5%	12%	3%

Fig. 11: Training and testing accuracy of our pipeline. First two rows show the statistic of our hand-labeled data (ground-truth). For the pipeline, we consider an app dual-use if LR classifier's confidence is more than 0.7 or if the confidence is within [0.3, 0.7] and majority of MTurk worker labeled it as 'dual-use'.

We repeated the scraping process in November for three days, following the publication of an article by The Daily Beast accusing Google of showing ads about illegal spyware [21]. We observed a total of 2,866 ad impressions linking to 186 domains, resulting from 432 search terms. Some searches yielded as many as 7 ad impressions on the first page of search results. The most advertised domain remained the same. We ran one further scrape in March for one day and collected 1,843 ad impressions linked to 137 domains and 372 search terms.

We analyzed all 96 domains that appeared in at least 10 ad impressions across all measurement periods. These 96 domains are associated with 11,831 ad impressions (95%). Of these domains, 20 belong to services offering public record or reverse phone number lookups. Those represent half (6,217) of the ad impressions. Another 22 domains are of tracking apps and software and account for 3,128 ad impressions. Eighteen domains (linked to 1,162 ads) belong to miscellaneous but relevant sites, including: manufacturers of physical tracking beacons, private eye services, blogs and forums of the kind discussed below, and social networking sites which facilitate infidelity. The remaining 34 domains linked to 1,324 ads are not at all relevant to IPS.

We analyzed the 598 search terms that returned ads across all measurement periods. We determined whether each term explicitly indicated that the searcher intended to engage in IPS. Terms that indicated the intent to track a cell phone but did not indicate that it was another person's phone (such as "best free

gps phone tracking app") or that indicated the intent to track a child's phone (such as "free family tracker app") were labeled "relevant" but not explicit. Terms that discussed a spouse but did not mention tracking (such as "cheating spouse forum") were also marked relevant but not explicit. Of the 598 search terms, 135 were explicit, 324 were relevant to IPS but not explicit, and 139 were irrelevant (e.g., "Spyro the Dragon"). Of 12,484 observed ad impressions, 58% were on explicit terms, 39% on relevant terms and 3% on irrelevant terms.

We further examined the 3,128 ad impressions shown for the 22 domains that sold IPS-usable software. Of these, 1,203 (38%) were shown on IPS-explicit terms, 1,920 were shown on IPS-relevant, but not explicit, terms, and four were shown on irrelevant terms. The rate of ads on IPS-explicit terms for specific apps ranged from 0%, in the case of TeenSafe, (the one app that admonished us when speaking to customer service, 590 total ad impressions) to 91% in the case of RemoteCellSpy (418 total ads). Though further study is required to find out which words in our search term is triggering the ad, the discrepancy in the number of IPS-explicit terms showing ads for one company but not for another seems to indicate that some companies are actively trying to advertise for IPS use.