Comparing User Perceptions of Anti-Stalkerware Apps with the Technical Reality

Matthias Fassl
CISPA Helmholtz Center for Information Security

Simon Anell
CISPA Helmholtz Center for Information Security

Sabine Houy
Umeå University

Martina Lindorfer
TU Wien

Katharina Krombholz
CISPA Helmholtz Center for Information Security

Abstract

Every year an increasing number of users face stalkerware on their phones [84]. Many of them are victims of intimate partner surveillance (IPS) who are unsure how to identify or remove stalkerware from their phones [49]. An intuitive approach would be to choose anti-stalkerware from the app store. However, a mismatch between user expectations and the technical capabilities can produce an illusion of security and risk compensation behavior (i.e., the Peltzmann effect).

We compare users' perceptions of anti-stalkerware with the technical reality. First, we applied thematic analysis to app reviews to analyze user perceptions. Then, we performed a cognitive walkthrough of two prominent anti-stalkerware apps available on the Google Play Store and reverse-engineered them to understand their detection features.

Our results suggest that users base their trust on the look and feel of the app, the number and type of alerts, and the apps' affordances. We also found that app capabilities do not correspond to the users' perceptions and expectations, impacting their practical effectiveness. We discuss different stakeholders' options to remedy these challenges and better align user perceptions with the technical reality.

1 Introduction

About one in five adults and even more young adults engage in snooping attacks on others' phones [54]. *Intimate partner surveillance* (IPS) is a specific subset of these attacks [13, 88]. Tool-based IPS often involves a type of spyware, called stalkerware (or surveillanceware), to collect live location data, contacts, call history, and text messages [15, 80]. According to the Coalition Against Stalkerware [84], 67,500 mobile users were confronted with stalkerware in 2019, a 67% increase compared to the year before. Randall et al. [76] estimated that at least 5,758 people in the US were targeted by overt stalkerware from March to May 2020. Two of the 22 apps they studied were available in the Google Play Store, the remainder were only available from third parties. In October 2020, Google banned surveillance apps from their store [37] and now only allows surveillance in parental control and enterprise management apps if they do not hide or obfuscate their surveillance practices. Hence, stalkerware often rebrands itself as parental control apps or moves to third-party websites. Most stalkerware occurrences in *clinical computer security* [43] consultations comprise such "dual-use" apps [15].

An analysis of online domestic abuse forums and an assessment of the stalkerware application (app) industry identified that IPS survivors are unsure how to recognize and remove stalkerware [49, 66]. Installing anti-stalkerware apps from the Google Play Store is one possible approach. Users may choose from various apps, ranging from traditional anti-virus companies offering general mobile security solutions to specialized apps detecting stalkerware and other spyware. Prices vary widely, some are as cheap as €5 (or $), but in-app purchase prices up to and beyond €100 (or $) are not uncommon. However, these apps come with severe limitations on Android since they often operate with simple name-based blocklists, which stalkerware can circumvent easily [10]. More worryingly, there have also been instances of fake anti-virus apps in the Google Play Store with limited to no functionality at all [22, 45, 63, 97]. Thus, the marketed promise of identifying stalkerware is at odds with many of these apps' abilities, constituting an expectation-ability gap. This problem affects users' ability to make informed decisions. Survivors should be made aware of these problems to allow them to question their reliance on them.

We conduct an exploratory case study with two anti-stalkerware apps to understand this mismatch between expectations and abilities. We focus on the following research questions: (RQ1) *What are the differences between users'*

Copyright is held by the author/owner. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee.
USENIX Symposium on Usable Privacy and Security (SOUPS) 2022.
August 7–9, 2022, Boston, MA, United States.

security perceptions and the anti-stalkerware apps' abilities?; and (RQ2) *How could research and design begin to remedy this mismatch and foster users' anti-stalkerware decisions?* We apply thematic analysis to app-store reviews to study perceptions of these apps. We also perform a cognitive walkthrough of the respective apps and then reverse engineer them to understand how their detection mechanisms work. Hence, we elicit expectation-reality mismatches by combining qualitative user research with a reverse-engineering approach. Based on app reviews, we identified five user approaches to building confidence in their anti-stalkerware choice, all of them intuitive to apply and with some degree of legitimacy. However, contrasting these approaches with the cognitive walkthrough and reverse engineering results demonstrates that they fail to inform users about apps' abilities to mitigate violence, abuse, and harassment. Our work helps improve the current state of anti-stalkerware by suggesting design directions, proposing toolkit-supported user decisions, and discussing systemic, platform-level approaches to combating intimate partner surveillance.

2 Background and Related Work

This section describes background information and prior work on intimate partner surveillance and our methodology.

2.1 Intimate Partner Surveillance

Insiders, i.e., persons who are familiar to the victims, are a threat to smartphone users that security experts underestimated in the past [60]. Insiders' access to victims' devices varies significantly. However, according to one study in the US, 31% of participants looked through others' smartphones without their permission [54]. Surveillance among intimate partners, a specific insider attack, is usually technically unsophisticated and relies on UI-bound attacks or ready-made apps [27]. Bellini et al. [13] and Tseng et al. [88] analyzed stories on online forums about sexual infidelity. Abusers justify their surveillance with their suspicion of sexual infidelity. They want to collect evidence, understand behavior, and control behavior [13]. Bellini et al. [13] identified a four-stage abuse cycle: setting the abusers' expectations, attitude change, escalation, and reflection. Tseng et al. [88] categorized IPS attacks based on physical and non-physical access requirements. They found that online communities are a good source of IPS threat intelligence because their users collaborate to create new IPS attacks.

Chatterjee et al. [15] identified apps that are dangerous in the IPS context. They found explicit spyware apps and more subtle dual-use apps with legitimate use-cases (e.g., FindMyFriend). Often, anti-spyware does not identify the latter as a threat. Parental control apps, a classic example of dual-use, also suffer from other privacy issues, e.g., collecting sensitive data and distributing it to third parties without consent [24]. To understand the "creepware" ecosystem, Roundy et al. [80] developed the *creeprank* algorithm based on guilt by association. As a result, hundreds of apps were removed from official app stores and presumably moved to third-party repositories.

Based on survivors' stories, Matthews et al. [57] identified different phases of separation and technology use. Survivors' safety in the "life apart" phase depends on identifying stalkerware. Havron et al. [43] and Freed et al. [26] created a computer security clinic for IPS survivors who readily accepted support in this format. However, since anti-stalkerware apps have a low barrier for entry, survivors presumably also use them as part of their protection ensemble. Lee et al. [48] extended the theory of planned behavior to understand factors leading to anti-spyware software adoption.

2.2 Users' Security Behavior

Due to a lack of structured security education, users learn their security behaviors haphazardly from various sources. Media, negative experiences, family, peers, workplace, IT professionals, and service providers are common advice sources [79]. However, all these sources focus on different aspects of threats [72]. Hence, no single source is sufficient. Giving security advice to individuals in situations of abuse is especially sensitive: affirmative steps to prevent attackers' data access suggest a lack of trust and may worsen abuse situations [50]. Emms et al. [23] suggested approaches to improve survivors' ability to avoid traces in ongoing abuse situations. Anti-stalkerware apps not specifically adapted for use in abuse situations may only be safe to use in the life apart phase. IPS survivors seek help and support in online forums from other survivors [49]. However, forum users often lack appropriate technical know-how, making it hard to recommend safe and effective anti-stalkerware apps. Reviews influence online consumer decisions in general. The quality of the review contents and the ranking affects consumer decisions more than the number of reviews and the sources' credibility [25]. Reviews can also influence security decisions, e.g., some users check app-store reviews before their update decision [87]. Most people also learn security lessons from family members' and friends' stories [73]. However, the stories' contents, the location, and the storyteller influence lessons' effectiveness. Social influence from peers affects security features' adoption, depending on the features' visibility to others [18]. Luca et al. [19] identified peer pressure from friends as the main factor for secure instant messenger adoption. Personal negative experiences also influence future security decisions. Vaniea et al. [89] found that users avoid updating after bad update experiences.

However, advice is not the only source of behavior – software prompts and automated security decisions also impact users' security behavior [78]. Mathiasen et al. [55] found that behaving securely does not necessarily result in a secure experience. According to them, careful design focused on creating secure experiences can increase security feature adoption. Distler et al. [21] found that including security-

related information in an e-voting process improved users' secure experience. They discuss how quick and smooth security mechanisms may impede users' secure experience despite improved usability—an idea they extend on in a framework of security-enhancing friction [20]. Users' mental models of potential attackers impact their adopted protection behavior [92] since each class of attacks calls for different protection mechanisms. Zou et al. [99] studied users' reasons for adopting and abandoning security and privacy behaviors. They found low adoption for recurring interaction practices and higher privacy practice adoption rates among low-income participants. Users abandoned security and privacy practices when they found them impractical, no longer saw their value, or perceived diminished risk. Similarly, users turn off protective measures such as firewalls when they find them complicated [75].

2.3 Review Mining and Analysis

App-store reviews inform users about the apps' quality, but also developers about bugs and feature requests, as well as researchers to gain detailed insights about apps. In light of the sheer number, informality, and shortness of these reviews, researchers either mine reviews to get a broad overview or use thematic analysis to examine a subsample in rich detail.

The software engineering community explored automated ways to mine user reviews for actionable development feedback. Prior work discussed several different automatic approaches to identify informative complaints in app reviews [16, 30, 53, 69]. Khalid et al. [47] used manual qualitative analysis to identify complaints about iOS apps.

Others have focused on automatically retrieving feature requests from reviews [46, 53] using natural language processing, sentiment analysis, and LDA models. Automatic analysis of app reviews can also inform developers about usability and user experience issues [12, 44, 58, 64]. Gu et al. [38] and Guzman et al. [39] applied sentiment analysis to understand how users feel about apps and individual features.

Researchers have also used reviews to study security- and privacy-related aspects of apps. Ha et al. [40] manually coded reviews to look for security and privacy complaints and found that about 1% of them concerned app permissions. Nguyen et al. [61] analyzed reviews for security- and privacy-related reports and traced 61% of security and privacy updates to corresponding user reviews. Voskobojnikov et al. [91] analyzed cryptocurrency wallets' reviews to understand security- and privacy-relevant UX issues. They identified a subsample of relevant reviews using machine learning and natural language processing and then applied thematic analysis. Gosh et al. [32, 33] qualitatively analyzed reviews of parental control apps to understand how children responded to them. They used a keyword search to filter children's reviews and applied thematic analysis. Children found the apps overly restrictive and privacy-invasive. They criticized their parents' reliance on these apps as a bad parenting technique.

2.4 Spyware Detection

In general, there are two basic approaches to detecting and analyzing malware, including stalkerware: static and dynamic analysis [5]. Static analysis is the understanding of a program at the syntactic source code or binary level [31]. Dynamic analysis focuses on an app's run-time behavior, including system calls and network traffic. For this purpose, researchers execute and observe apps in controlled environments [52].

Knowing the reliability of on-device anti-malware scanners (commonly referred to as anti-virus) is crucial for end users' safety. These scanners base their detection mechanisms on either static or dynamic analysis. However, compared to security solutions on desktop operating systems, mobile security apps have limited visibility into other apps due to extensive sandboxing, rendering behavioral heuristics unfeasible [17, 51, 70, 71]. Security solutions thus have to rely on signatures based on code-level characteristics or use machine learning [9, 51]. Related work has investigated in-depth how easy it is to evade those signatures [11, 41, 71, 77, 98]. Yet, no study so far compared the robustness of detection mechanisms to the trust users put into these security solutions.

3 Methodology

We explore the gap between users' expectations of the apps' functionality and the apps' technical abilities. Understanding this mismatch helps to improve users' protection against stalkerware. First, we apply thematic analysis [14] to app-store reviews of the two case-study apps to understand users' security perceptions and expectations. Based on the resulting themes, we perform cognitive walkthroughs of the apps and analyze them to understand how they detect stalkerware.

3.1 Selection of Anti-Stalkerware Apps

Spyware poses an increased danger to Android users compared to iPhone users [42, 66]. Apple's iOS claims tighter security controls [7] and does not allow apps with "functionality it does not actually offer (e.g., iOS-based virus and malware scanners)" [8]. Hence, we focus on Android apps.

To cover a variety of app abilities and user expectations in our qualitative analysis, we base our selection on Chatterjee et al.'s anti-spyware list [15]. From the most-downloaded anti-stalkerware apps, we chose two to perform static analysis on: Mobile Security, Antivirus & Cleaner by Lookout[1] (100M+ installs) [86]. From the long-tail, we read app-store pages and chose a data-rich example suitable for further qualitative analysis: Anti Spy Mobile PRO[2] (100k+ installs) [85].

Fraudulent reviews and manipulated ratings plague free apps [74, 95, 96]. Therefore, we prefer to analyze reviews of paid apps. Lookout Mobile Security is free to download on the Google Play Store and uses an in-app subscription model.

[1] Version: 10.33-6652654, Downloaded: June 2020
[2] Version: 1.9.10.51, Downloaded: June 2020

We can not differentiate between subscribed and unsubscribed users' reviews. Hence, we also analyzed reviews from unsubscribed users. Lookout Mobile Security is more extensive and complex than Anti Spy Mobile PRO. Lookout Mobile Security markets itself as a fully-fledged security solution, with anti-spyware as only one of its features. In contrast, Anti Spy Mobile is available as a free or paid version (€ 4.90 or $ 3.99) on the Google Play Store. The only difference is that the paid version has automatic daily background scans. We only analyzed the paid version's reviews.

The focus on these two apps affects the results twofold: First, their features are not representative of all security apps marketed as anti-stalkerware. Second, Lookout Mobile is preinstalled for some users, so the lack of choice may impact users' reviews. Hence, reviewers' sentiments from these two apps are not generalizable to all security apps that market themselves as anti-stalkerware.

3.2 Analysis of App-Store Reviews

To understand how users perceive our case study's anti-stalkerware apps and engender trust in them, we applied thematic analysis [14] to a sample of their app-store reviews.

We fetched all reviews from the Google Play Store.[3] We randomly sampled 200 comments from each app in German and English, languages all involved researchers understand well. To ensure the reviews had enough content, we only considered comments with at least ten words. We analyzed a total of 400 reviews for Lookout. Anti Spy Mobile PRO, had less than 200 reviews fulfilling our criteria, so we analyzed a total of 13 German and 102 English reviews for this app.

At the start of the thematic analysis, one researcher read all reviews and created an initial codebook. With it, both researchers coded the entire review sample. During the coding procedure, both researchers kept notes on potential themes in the data. This resulted in an inter-coder agreement of Krippendorff's alpha $\alpha = 0.86$, which suggests *excellent* agreement. Afterward, the researchers discussed all mismatches and the themes they identified. Vague reviews with multiple valid interpretations caused most of the disagreements. Resolving conflicts increased Krippendorff's alpha to $\alpha = 0.98$. Table 1 in the Appendix presents the initial codebook.

The discussions led both researchers to agree on a focus on safety and security perceptions. We repeated the above procedure and constructed an additional codebook. Krippendorff's alpha was $\alpha = 0.78$ after the initial round of coding, suggesting *substantial* inter-coder agreement. Discussing all mismatches increased Krippendorff's alpha to $\alpha = 0.96$. At the start of the discussion, the researchers added a "time of experience" code and applied it whenever appropriate. Table 2 in the Appendix presents the revised codebook. Afterward, both researchers discussed the identified themes and the presentation of the results.

[3] Anonymized JavaScript code: https://pastebin.com/bRZ1v0XS

3.3 Anti-Stalkerwares' Technical Abilities

After identifying security perceptions and expectations, we used *theoretical sampling* to understand these apps' technical abilities. Thus, we collected data about the user interface and the apps' internal detection mechanisms.

We conducted cognitive walkthroughs for both apps to improve our understanding of the reviews focusing on user experience. Based on the previously discovered themes, we focused on the following: (1) method of invoking scans (manual, scheduled, event-triggered), (2) type and amount of information in reports, (3) false positives in a general use scenario, (4) visible user interactions under regular usage. We screenshot these parts of the case study apps and deductively code them with the codebook from the review analysis.

Additionally, we reverse-engineered the case study apps to understand how they detect stalkerware. In both cases, we started with *static analysis*, i.e., decompiling and inspecting their source code. We used *dynamic analysis* to verify the results and to understand run-time behavior. This allowed us to observe and inspect the output of the apps' scanning and evaluation functions for potentially harmful behavior.

3.4 User Perceptions vs. App Capabilities

Finally, we juxtapose the trustworthiness and security perceptions with theoretical samples from each case-study app to point out mismatches between perceptions and technical reality. As far as possible, we embed the perceptions and theoretical samples into related work to provide an additional broader context. We evaluate the benefits and drawbacks of users' strategies for choosing anti-stalkerware.

3.5 Ethical and Legal Considerations

Using public data for research without explicit consent is an ethical challenge, especially concerning intimate partner abuse. Even though users can remove their public reviews, we handle all data with care to minimize potential harm. We omit usernames and rephrase quotes if they contain hints of abusive behavior, rendering identification difficult.

Reverse engineering is a legal grey area. In the US, good-faith security research is exempt from copyright law and the DMCA [65]. In the EU, decompilation is explicitly allowed to ensure interoperability with other software [90]. EU copyright law only protects the concrete expression of the source code, not the underlying ideas and principles. We carefully reviewed our results to avoid publishing information that could be considered a concrete expression.

We want to minimize potential harm from publishing results of our technical analysis. After a careful review, we identified three types of potentially harmful information: (1) well-known stalkerware that apps do not identify correctly, (2) flawed general approaches to detecting stalkerware, and (3) specific implementation details about threat classification. We informed the app providers about well-known stalkerware

their app did not identify before publication. The general flaws we identified are well-known; existing spyware and state-of-the-art anti-spyware already take them into account. Hence, publishing these general flaws does not introduce new harm. Specific implementation details on how apps classify threats are out of scope for this work. Since stalkerware could use these findings to evade detection, we refrain from publishing them. Our institution's ethical review board (ERB) approved this study.

4 Users' Perceptions of Anti-Stalkerware

To understand how users perceive the security of anti-stalkerware apps, we analyzed the app-store reviews of the two apps in our case study. We included a total of 518 reviews in our study and performed thematic analysis to find higher-level themes and patterns in the data. In the following, we report the results from this analysis, i.e., our findings on users' approaches to engendering trust in anti-stalkerware apps, general observations, and contradicting user expectations.

We identified five approaches users apply to convince others of anti-stalkerware apps' usefulness and trustworthiness.

Potentially harmful incidents. First-hand experience of an apps' protection is a popular way for users to establish trust. This approach to establishing trust covers a variety of different features. Amongst others, we have found praise for adware detection, e.g., *"has already found and removed adware three times." (R326)*, spyware detection, e.g., *"Someone had put a tracking app on my phone [...] I had it figured out in about 10 minutes!" (R425)*, and theft prevention, e.g., *"It [...] has saved me from losing my phone not once but twice to thifes." (R132)*. Interestingly, reviewers did not seem concerned about apps' potential shortcomings in other areas. One great first-hand experience may suffice to convince users of an app's general effectiveness.

However, we also observed this effect the other way around. As soon as users have negative experiences with core features, they lose confidence. In one case, the reviewer knew that an ex-partner spied on them, but the anti-stalkerware did not detect any malicious app: *"Never purchase this! My ex is still reads my messages - it's a disgrace" (R477)*. Similarly, this reviewer's trust vanished as soon as they realized they could not locate their stolen phone: *"The whole reason I have this app is in case I lose my phone." (R069)*.

While effective security apps must protect users in cases of attacks, a single thwarted attack is not a good indicator of a security app's effectiveness.

Reassuring user experience. Security apps' user experience influences the users' opinions about these apps. Frequent reminders of threats, updates, or scheduled scans keep users informed about the app's activity. Generally, attacks on users' security will be rare. So that these reminders of the ongoing protection effort can add a feeling of security for users: *"Get notified my phone is secure. That makes me feel better." (R165)*.

Other users may see these reminders as a disruption of their regular phone use, e.g., *"the notification is permanently visible in the status bar. This is unsettling and annoying." (R202)*.

For security use-cases, where apps might only rarely need to intervene, reassuring user experience is necessary to communicate that the app is still there and doing its job. However, reassuring user experience is independent of actual security. Hence, app developers may misuse this concept.

Building trust over time. Frequently, the history of app use influenced trust. Similar to human relationships, using the app over an extended period reassured users and increased their trust in the security app. We found three types of time references: establishing authority by stating experience, insufficient evidence of protection over time, and satisfaction with the absence of incidents.

In case of establishing authority, reviewers usually said they had used the app for years before telling us their verdict, e.g., *"Works as advertised have used it for years" (R173)*. Some reviewers expected security apps to demonstrate their effectiveness. R476 assumed the app was a scam because they could not determine what it does: *"I cannot tell that this does anything for my phone so I think this is a rip off"*. However, other reviewers were happy and felt safer when the security app did not find anything: *"Haven't found anything yet but thats a good thing!! Feeling alot more safe." (R475)*

These contradicting positions are interesting since they demonstrate two fundamental ways users think about apps' security. In the first one, users demand evidence of functionality, even if there is nothing wrong with their smartphone. The other approach assumes the security app's effectiveness without evidence. Even though both reviewers used the same app, they ended up with different trust assessments.

Testing app's abilities. Numerous users did not wait for incidents in their day-to-day life to establish trust. They decided to test the apps' abilities. They compared the abilities of different anti-stalkerware apps, e.g., *"This app missed two spyware apps that the others detected." (R470)*. Some knew they had spyware installed and checked if a particular anti-stalkerware could remove it: *"Can't find the spyware that is obviously installed on my phone." (R512)* R291 reported using an EICAR test file to check if the security app would detect it: *"Garbage. Eicar test antivitus not detected" (R291)*. In this case, the reviewer successfully tested the 'lost phone' feature: *"Locating/Alarm etc always worked when tested" (R344)*.

In general, testing security features is a solid way to build trust. However, comprehensively testing apps' malware detection abilities is hard. Other security features, such as the 'lost phone' feature are easier to test than malware detection's effectiveness. Hence, reviewers could have a misleading impression of their app's abilities even after testing them.

Third-party recommendations. Reviews rarely referred to third-party resources to justify their trust in anti-stalkerware apps. In one case, a friend in IT security recommended an app: *"My friend who is in IT security suggested this app to me" (R131)* In another case, a reviewer referred to a study: *"saw a study that showed this had best spyware detection rate (but also false positives)" (R423).*

Users who got anti-stalkerware recommendations from third parties have delegated trust establishment. For them, the user experience of a security app is not as crucial as for other users – they are already confident in its security.

4.1 Observations

During our analysis, we also observed other noteworthy trends among the reviews: emotional language, assemblages of security tools, and cases of tracking family members.

We found that reviewers often used emotionally loaded language. Positive reviews, such as R145, described the protection app as a sort of guardian angel: *"It's a guardian keeping an eye on my stuff"*. The name of one of the apps in our case study, i.e., Lookout, might explain why reviewers make this connection. Negative reviews often used strong language when talking about the apps' shortcomings. Such as R114, who complained about the app's malware detection ability: *"Pathetic virus support"*, or R014, who just wanted to remove the app altogether: *"take this Crappy off [my phone]"*. However, since app-store reviews are voluntary, these observations could be due to self-selection bias, i.e., users who feel betrayed or well protected by the app submit more reviews.

Some reviewers did not evaluate the app independently from others. Instead, they considered how the app fits into their assemblage of security tools, e.g., *"Nice addition to any security set up." (R402)* or *"Lookout (Basic license) is good pair with Avast Mobile Security and CCleaner." (R098)*. In such cases, users focus less on a specific tool's efficacy but rather on the feature set of the entire assemblage. However, some of these tools expect to be standalone tools, which may impact the resulting user experience.

One reviewer explicitly described their use-case for the app as tracking family members. *"We did not change anything but whenever I try locating my son there is an error." (R155)* We assume that parents such as these have only the best intentions for their children's safety. However, Gosh et al. [32] found that affected children perceive their parents' surveillance as overly restrictive and privacy-invasive. Our case also illustrates how users employ security apps to subvert their intended use-case.

4.2 Contradicting User Expectations

We found two approaches to trusting the apps in our case study: (1) trust, based on absent negative experiences with the app, and (2) no trust without proof that the app works as intended. Using the first approach increases trust in the security app the longer it runs without incidents. Users employing the second approach either wait until the app detects an issue or challenge the app to trigger an alert. R260 is exemplary for the first approach: *"I have had this app on all my devices over the years and no problems of any kind"* R215 is an example of the second approach: *"I've not had any positive hits from this yet, so it's difficult to say how good or bad the app is."*

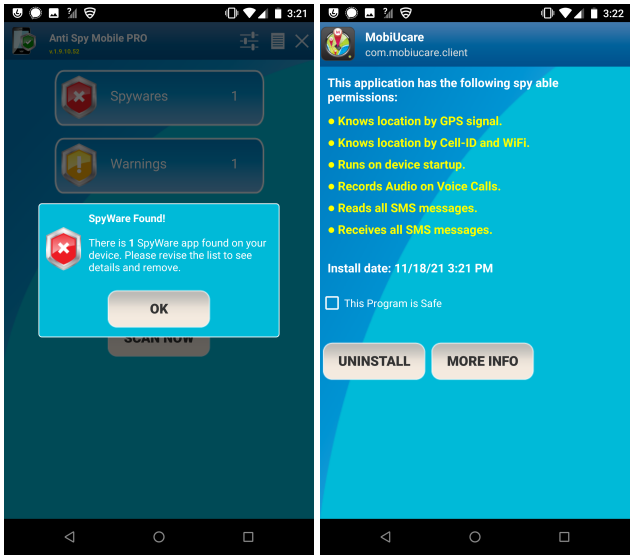

Figure 1: Anti Spy Mobile PRO's response to a well-known spyware app.

The app's user interaction impacted users' trust in two contradictory ways: Some users thought the app was not doing anything when they could not observe any user interaction with it, i.e., they felt reassured by visible UI elements. Others interpreted the missing user interaction as a security indicator, expecting the app to respond only to security issues. R121 feels reassured when Lookout communicates that it is working: *"it lets me know they are working by updating me at various time intervals and pops up on your screen when you are not thinking about them" (R121)* R250 would feel more protected if Anti Spy Mobile were to indicate its ongoing operation: *"there should be an anti-spy guard for the icon on the home screen. That would enhance users to feel protected and safer" (R250)* In contrast, R065 is happy that the app stays silent and in the background: *"it silently keeps my phone in check from the behind the curtain" (R065)*

5 UI Walkthrough of Anti-Stalkerware

During our thematic analysis of the app-store reviews, we identified two approaches on how users establish trust with anti-stalkerware apps based on their user interface: (1) Incidents with potential harm and experiencing how the app handles the situation builds users' trust; (2) Apart from potentially harmful incidents, users appreciate anti-stalkerware's reassuring security experience during everyday use.

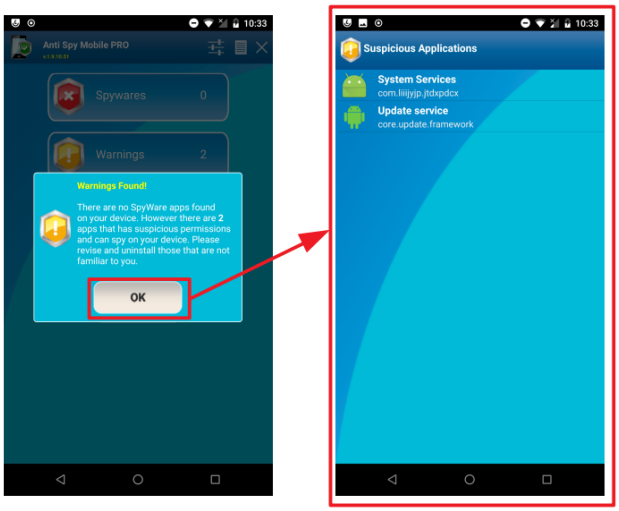

Figure 2: Anti Spy Mobile PRO's response to spyware apps that are not on its list of well-known spyware.

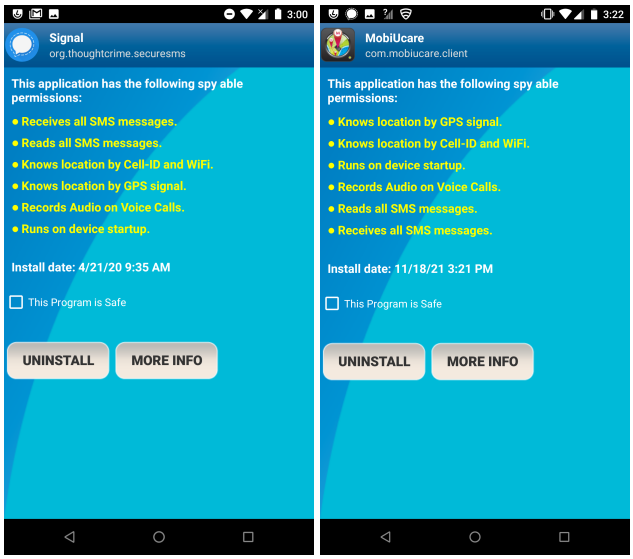

Figure 3: Additional information provided by Anti Spy Mobile PRO on a suspicious app on the left and a well-known spyware app on the right – both apps request the same "spy able" permissions.

This section reports the results of a cognitive walkthrough [82, 93] focused on these two trust establishment approaches. For the purpose of this walkthrough, we assumed that malicious parties may have had direct access to the phone before, but they no longer do at this point. When malicious parties still have direct access, removing electronic traces of anti-stalkerware usage afterwards is necessary to keep its users safe [23]. We simulated harmful incidents by installing several spyware apps on a smartphone that we reserved for this purpose. In the resulting user interactions, we document and inspect all the parts of the UI flow and answer guiding questions about the effect on users' trust. We simulated the day-to-day experience by using the smartphone with the installed case-study apps for 48 hours as our regular phone. We browse the web, download data, and install apps. We document and inspect user interaction and answer guiding questions about the effect on users' trust.

5.1 Potentially Harmful Incidents

Anti Spy Mobile PRO. Opening the app shows three different classifications of apps (as buttons): (1) *Spywares* for well-known blocklisted spyware apps; (2) *Warnings* for all suspicious apps not on the blocklist; (3) *All Applications* for all other apps.

Anti Spy Mobile automatically starts a scan when users open the app for the first time. Users may trigger a scan manually with the *Scan now* button or enable automatic daily scanning in the preferences (which is the default setting). After each scan, a dialog box presents the number of identified well-known spyware apps. If it did not find any, it presents the number of suspicious apps instead. Confirming the dialog box brings users to review the apps in question (as seen in Figure 1 and 2).

To test Anti Spy Mobile's reaction to a well-known spyware app, we installed MobiUcare (Phone Locator) on our test phone. Figure 1 shows the resulting *"SpyWare found"* dialog. After confirmation, Anti Spy Mobile shows the name, privacy-infringing permissions, and installations date of the detected spyware app. The *"More Info"* button would usually lead to the corresponding listing in the Google Play Store. However, this results in an error message since this app's removal from the store.

We installed two more spyware apps: mSpy Cellphone Tracker and SpyFone. The FTC banned the latter in September of 2021 [29]. Figure 2 shows that it does not consider them well-known spyware. Instead, it informs users about suspicious apps on their phones. The text describes the classification based on requested permissions and suggests how to deal with these apps: *"you should take a close look at them and uninstall them if you are not familiar with their existence"*.

Selecting suspicious apps reveals more detailed information about them (Figure 3), such as their name, suspicious permissions, and time of installation. This view offers users three responses. First, users may want more information about the app in question. However, the corresponding button leads to the Google Play Store website, which may not provide users with sufficient threat information. With MobiUcare, the button generates an error since the app is no longer on the app store. Second, users can uninstall the app directly with a button click. However, if it concerns admin apps, this results in an error message: *"Uninstalling MobiUcare unsuccessful"*. The app does not provide any guidance in this case and acts as if the user never pressed the button in the first place. Third,

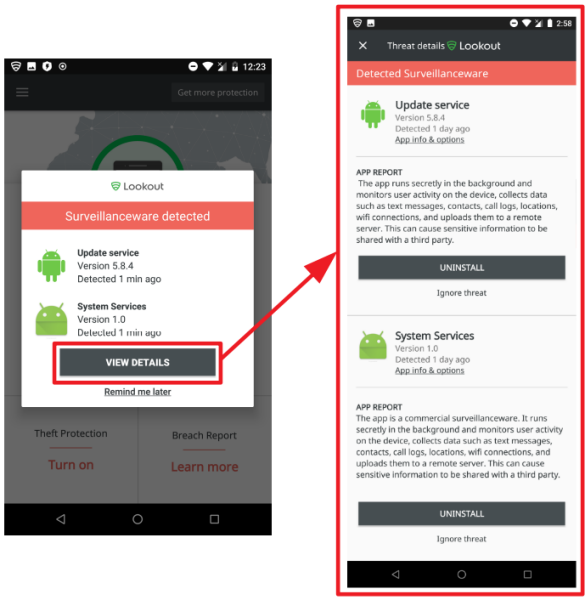

Figure 4: Lookout Mobile Security's scan results identifying the spyware apps as surveillanceware.

if users do not want to take further action, they mark the app in question as *"safe"*. Then Anti Spy Mobile will stop notifying them about the app. Figure 3 shows that the threat response interface is independent of the identified threat. Anti Spy Mobile treats apps with merely suspicious permissions (the Signal messenger in this case) in the same way as apps on its list of well-known spyware.

Lookout Mobile Security. Lookout Mobile automatically scans all installed apps after installation. Users can start a scan manually at any time (see Figure 5).

To test Lookout's response to stalkerware, we installed MobiUcare, mSpy Cellphone Tracker, and SpyFone. Lookout Mobile correctly identified all three and classified them as *Surveillanceware*. In Figure 4 a pop-up window shows all identified apps with the option to either view details or set a reminder. The remind later option does not require users to specify a time that works better for them. Such commitment devices can increase security compliance [28].

In the detailed overview, Lookout shows a classification (e.g., Surveillanceware), logo, name, version, time of detection, and an app report for each identified threat. Reports comprise three parts: a statement if the app is a commercial surveillanceware (if applicable), a list of human-readable permissions, and a generic explanation about third parties monitoring user activity without consent. The only context-dependent information seems to be Lookout's analysis if the app in question is commercial surveillanceware.

Lookout affords users three responses for detected threats. First, users may click on App Info & options, leading them to the system's overview of the app in question. Second, a highlighted uninstall button. While Lookout does not explicitly

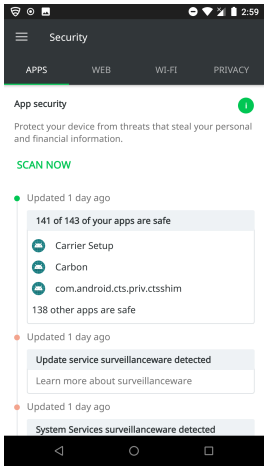

Figure 5: Dashboard of Lookout Mobile Security with scan history and re-scan option.

suggest an appropriate response to the threat, the highlighted button strongly suggests uninstalling. Lastly, it offers the option to ignore threats. Lookout does not provide users an explicit discussion of these options, not even when it identifies commercial surveillanceware.

Additionally, users have access to the scan history (see Figure 5). Upon detection of surveillanceware, this view offers users to "learn more about surveillanceware", leading them to the built-in threat encyclopedia. The encyclopedia provides a general overview of surveillanceware abilities and only mentions a vague threat model, i.e., "Surveillanceware apps are typically installed directly by someone with physical access to the target device". The encyclopedia avoids discussing appropriate user responses.

5.2 Reassuring Everyday Experience

Anti Spy Mobile PRO. Apart from manual scans in the app itself, Anti Spy Mobile barely interacts with users. The paid version automatically scans all apps and notifies users about the results once per day (see Figure 6). This notification does not warn about suspicious apps. Anti Spy Mobile does not intervene during day-to-day activities, such as browsing the web, downloading files, or installing apps (from the Google Play Store or third-party repositories).

Lookout Mobile Security. In general, Lookout Mobile focuses on reassuring user interaction. A sticky icon in the status bar and a permanent notification (shown in Figure 7) informs users that Lookout is active and that "everything is OK". Another aspect of Lookout's user interaction is its reactivity to the users' actions. It warns users about malicious files or apps immediately after downloading or installing them, respectively. Additionally, Lookout has a setting to notify users about a WiFi network's safety at connection time. Immediate responses improve users' mental models when the notification links causes and effects [83].

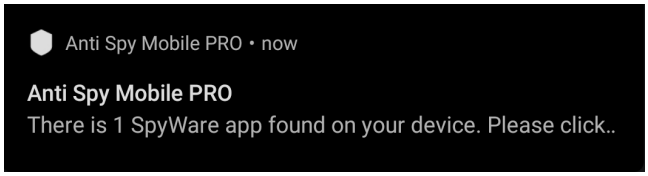

Figure 6: Anti Spy Mobile PRO's daily scan notification.

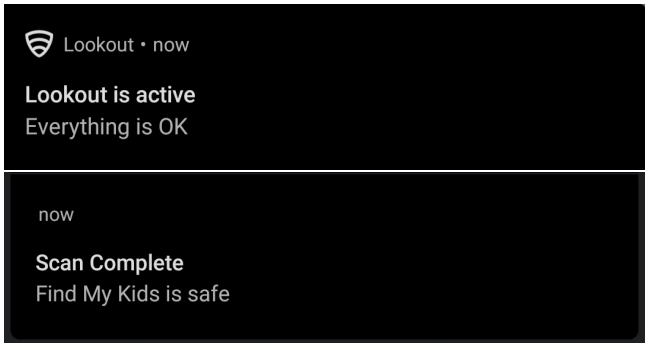

Figure 7: Lookout Mobile Security's reassuring notifications.

Enabling Lookout's VPN-based safe browsing feature did not affect the surfing experience. By default, Lookout analyzes downloaded files for threats (according to the description in the settings). Downloading regular files did not create a response from Lookout. However, it reacted when it detected spyware in a downloaded .apk file (Android Package, i.e., the Android app distribution format). Installing apps always created a response, regardless of the origin. Interestingly, Lookout considers the app Find My Kids safe (see Figure 7), while Anti Spy Mobile considers it well-known spyware.

6 Anti-Stalkerware under the Hood

Our thematic analysis identified two trust establishment approaches that users apply to anti-stalkerware. First, they build trust over time after seeing which threats the app caught and which it did not catch in time. Second, reviewers actively challenged the anti-stalkerware's abilities by installing known spyware on their phones. Both approaches are based on users' partially correct understanding of how to evaluate detection mechanisms.

To take a closer look at the detection mechanisms of our case-study apps and to understand how they determine which installed apps are threats, we performed static code analysis and dynamic run-time analysis. We follow established best practices (as outlined by OWASP [62]) for mobile app testing and rely on selected open-source tools. Android apps are typically written in Java, compiled to Dalvik bytecode, and then packaged as .apk files (essentially a zipped archive) [34]. A common first step is to transform this bytecode back into Java source code for easier comprehension. To do so, we use the Dalvik-to-Java decompiler `jadx` [3]. To monitor the run-time behavior of the case-study apps, we installed them on a Nexus 5 phone and instrumented them with Frida [2]. This tool allows reverse engineers to inject and execute JavaScript in the analyzed app. We use this feature to inspect the app's classes, methods, and data fields guided by the results of the static analysis. We further use the web proxy Fiddler [1] to intercept and inspect network traffic to the apps' backend server, if any.

Anti Spy Mobile PRO. We started by locating the main activity of the app, representing the UI shown to users when they first open an app. The class `AntiSpyActivity.java` represents this activity and loads the start screen defined in XML format (`/resources/res/layout/start.xml`). This screen contains the *Scan Now* button, which triggers the scanner activity (`ScannerService.java`). This activity implements the core functionality of Anti Spy Mobile PRO: it calls the Android PackageManager [36] to get the package names of all apps installed on the device and iterates over it.

The app distinguishes between two relevant types of installed apps: *SpyWare Applications* and *Suspicious Applications*. It identifies the first category by matching apps' package names against a list of well-known spyware apps. This blocklist of package names is embedded in the app as an XML file (*blackListPackagesDefs* in `resources/res/values/arrays.xml`). For the second category, Anti Spy Mobile PRO retrieves the apps' requested permissions to check for "spyable" permissions related to location, microphone, and SMS access. If the sum of these weighted permissions exceeds a certain threshold, it flags an app as suspicious.

The XML file that contains the blocklist also contains an allowlist of package names (*whiteListPackagesDefs*) of apps that presumably would trigger false positives based on their permissions. This list contains for example different browsers, but interestingly also security solutions such as Lookout Mobile Security. In its current version, the blocklist contains 494 entries, while the allowlist contains 146 entries, with 30 of these package names matching apps available on the Google Play Store, respectively.

We reverse-engineered the free version (Anti Spy Mobile Basic) and confirmed that the only difference is the option to schedule automatic background scans.

We further executed Anti Spy Mobile PRO to confirm our findings from the static code analysis and inspect its behavior during the actual scanning process. During this experiment, the app classified neither of the two spyware apps mSpy and SpyFone as SpyWare because its blocklist does not include them. However, it classified them as suspicious based on their permissions.

Lookout Mobile Security. This app is more complex than Anti Spy Mobile PRO, both in terms of code and UI. In this case, we started by looking for the *Scan Now* button in the dashboard UI (see Figure 5). This button triggers a SQL query for the already stored results of the previous scans. We then looked at the code populating this database, which is split

across a number of different classes. We found that Lookout Mobile Security also collects information about each installed app from the Android PackageManager [36]. In addition, for apps classified as malicious, it also stores an assessment including the classification category, assessment ID, severity of the threat, and the response type.

The actual scanning mechanism is implemented both as a local and a cloud scan. In the case of a local scan, it checks for assessment in the `Policy.FLX`. This policy is distributed via over-the-air (OTA) updates, i.e., updates automatically pushed to the app without any active user interaction. For cloud scans, the app creates a request to `https://appintel.mobilethreat.net` with hashed information about the app under assessment.

Monitoring the network traffic of the app using Fiddler, we observed that during the first scan it received data from `https://ota.lookout.com`. We identified this as the source of the OTA policies, but could not identify its format. Thus, using Frida, we injected JavaScript into the process to inspect the list of assessments read from this policy file. Most of the assessments seem to be in the form of signature-based detection methods, i.e., as a blocklist. Lookout detected both spyware apps (mSpy and SpyFone) as surveillanceware based on this blocklist.

Comparison of detection mechanisms. Both Anti Spy Mobile PRO and Lookout Mobile Security detect mSpy and SpyFone, the spyware apps. However, the first app merely classifies the two spyware apps as suspicious, while the second one accurately recognizes both as surveillanceware.

Anti Spy Mobile PRO mainly works with a block- and allowlist of package names. However, package names are weak identifiers of Android apps. The Google Play Store uses it to uniquely identify apps and recommends following Java package naming convention, i.e., to "use Internet domain ownership as the basis for package names (in reverse to avoid conflicts with other developers" [35]). Still, developers can choose arbitrary or conflicting package names for their apps, particularly when they are distributed via third-party repositories. Malware authors have been known to use the tactic of imitating package names of benign apps, or randomly generating package names to evade detection [52]. The package names of mSpy (`core.update.framework`) and SpyFone (`com.rzjzmlrm.vhqpmgzo`) seem to follow this pattern. Technically, stalkerware distributors could even automatically generate new package names for each customer.

Furthermore, these lists are part of the resources embedded in the .apk file, and the app does not implement any functionality to update this file. Thus, any changes in the blocklist need to be pushed as part of app updates through the Google Play Store—which users may or may not install [59, 87]. The update history indeed includes *[UPDATE] Spyware definitions update*, but updates have been sparse since 2018 [6].

In addition to the detection based on the package name, Anti Spy Mobile PRO also flags apps as suspicious if they request permissions that could be used for spying. Nevertheless, Anti Spy Mobile PRO does not provide more information about these apps than the requested permissions to the users and does not describe or explain what these apps do.

Lookout Mobile Security, on the other hand, dynamically fetches signature-based blocklists from the server and checks for newer versions during each launch. However, in this case, the scan is a "black box": we have no insights about the type of scans performed on Lookout's servers and the features they base their detection on.

7 Discussion

We compare our thematic analysis results, i.e., users' strategies for establishing trust in their installed anti-stalkerware, with our user interface walkthrough and reverse engineering results – highlighting the expectation-ability gap. Then we discuss different stakeholders' options to reduce this gap and improve users' anti-stalkerware decisions in the future.

7.1 Contrasting Users' Expectations with Actual Protection Capabilities

Potentially harmful incidents. One of the ways reviewers decided to trust anti-stalkerware apps depends on their incident response. This approach relies on apps' ability to detect incidents. Users' trust depends on the information and user agency that apps provide. Our walkthrough revealed that Anti Spy Mobile PRO's suspicious apps produced easily identifiable false positives – potentially decreasing users' trust. Also, we found inconsistent results: Anti Spy Mobile considered *Find my Kids* well-known spyware, while Lookout Mobile considered it safe. This mismatch highlights the need for context-sensitive classification, especially for dual-use apps. Neither app did a great job informing users about specific threats and providing context-appropriate user agency options. For example, Anti Spy Mobile PRO offers the same information and response options, whether it concerns well-known spyware or merely suspicious apps. Reverse engineering the apps showed that Anti Spy Mobile PRO uses a package name list of well-known spyware apps and a list of well-known benign apps. Updating these lists requires an app updating the app. Lookout Mobile checks apps against local OTA policies, regularly updated from Lookout's servers. Anti Spy Mobile PRO further uses a permission-based approach to identify suspicious apps not on the list of well-known apps, resulting in easily identifiable false positives. Hence, relying on potentially harmful incidents as a strategy to establish trust with anti-stalkerware apps comes with risks. It relies on users' ability to recognize harmful incidents to understand if the app should have detected and prevented them. Waiting for such moments is risky. Ideally, users trust their anti-stalkerware app before they face attacks. Lastly, awarding trust in this way may deceive users. One instance where the app protected them may lead users to overgeneralize the assumed protection.

Reassuring user experience. The analyzed reviews contained praise for reassuring user interaction in benign everyday scenarios. In addition to the regular alerts in case of threats, Lookout Mobile incorporates user interface elements that communicate the current positive security status, e.g., "everything is OK". Showing users the security mechanisms during threats as well as in benign situations helps build users' mental models [83]. Distler et al.'s study [21] suggests that visualizing security mechanisms improves user experience. Notably, in our case study, Lookout Mobile always seemed confident in its safety assessments. In contrast, Anti Spy Mobile depends on permissions-based classification — leading to false positives. In addition, Lookout Mobile was very reactive, immediately notifying users about their actions' safety consequences. The timing of privacy and security notices may affect users' decisions in general [4]. Observing links between cause and effect forms users' mental models, making this immediacy between action and response beneficial [83]. However, moderately delayed privacy feedback may be a compromise to minimize interruption [67]. Reassuring user experiences have benefits in benign situations. They improve users' mental models and appear to improve user experience overall. The immediate response to potential threats may improve users' mental models by linking cause and effect. The certainty of anti-stalkerware's verdicts, warranted or not, may heighten users' trust. Ultimately, reassuring user experiences do not make apps more secure. Hence, users who rely on this trust establishment approach are prone to deception.

Assumptions about apps' detection capabilities. Reviews contained two approaches based on assumptions of the anti-stalkerware's detection abilities. First, reviewers evaluated the app's abilities over time, building trust similar to a personal relationship. Second, reviewers explicitly tested and challenged the app's detection ability with selected spyware or test viruses. Both approaches are flawed. Using the first approach, users assume they can detect a threat when the app can not. Since they may not recognize when the app fails to detect threats, they may only be aware of incidents where the app protects them. Using the second approach, users generalize their test results from a single test to the apps' abilities to detect other malicious software, which might seriously mislead users. Even worse, since they tested the apps' ability personally, they put significant trust in their assessment.

Reliance on third-party evaluations. Some reviewers exclusively relied on third-party evaluations of anti-stalkerware apps. Depending on the third party may be the safest choice to establish trust. However, it also comes with drawbacks. First and foremost, trust in the third party is required — moving the issue of trust establishment from the app to the third party. Then, the third party has to have reviewed the users' chosen app. The effectiveness of this approach relies on reputable third parties. Ideally, trusted third parties are well-known for providing fair assessments. However, social effects may impact the choice of trusted third parties. Users rely on tech-savvy family members and friends even when they can not provide fair assessments. In any case, users can not influence and may not even know which aspects third parties consider for their reviews (e.g., usability, user agency, detection rate).

Relying on third-party reviews, users do not experience how the app reacts in case of an incident, which may affect their comfort, comprehension, and ultimately their safety.

7.2 Implications and Future Work

The thematic analysis results suggest that judging anti-stalkerware apps' efficacy is hard for users. In the current circumstances, their safest option is to rely on IPV-specific evaluation results of certified antivirus testing labs. In the future, we should try to support and improve users' existing evaluation approaches and give them more agency to safely build trust in anti-stalkerware apps. However, adapting apps and operating systems to make intimate partner surveillance difficult and less surreptitious would likely limit the proliferation of stalkerware and other abuse-enabling apps more effectively.

Reassuring experiences are useful (if done correctly) but cannot be trusted. One of the themes in our thematic analysis was that users felt reassured and well protected based on UI elements. The UI walkthrough confirmed that one of the apps relied on positive messaging to communicate to users about its work. Mathiasen et al. [55, 56] refer to this as *secure experiences*, which are not necessarily the same as security. According to them, users will base their security decisions on previous secure experiences. Spero et al. [83] argue that user interfaces that hide security mechanisms hinder users from building detailed mental models of security. Hence, security mechanisms should present users with model-building information, whether they face security risks or not. As an example, Distler et al. [21] found that visualizing security mechanisms in an e-voting apps led to an increase in perceived security. While these kinds of reassuring and secure experiences may be understudied, they appear to provide several benefits: (1) they communicate to users that a security system is working, even when no security risk calls for action; (2) they may improve users mental model of security; and (3) they help improve users' security decisions later on. However, these kinds of secure experiences become a problem if they oversell the actual security, regardless of the intention. Therefore, simple reassurances that everything is safe may not be the best approach to building secure experiences. The anti-stalkerware apps in our case study probably use reassuring experiences to justify their existence to users. Without them, it may appear like anti-stalkerware apps do nothing of value, even when they work well. In summary, reassuring user experiences may improve users' mental models and security decisions, but users cannot rely on them alone to establish trust in security mechanisms.

Demonstrate stalkerware detection to users. In our thematic analysis, we found reviewers used several different (flawed) tactics to evaluate the detection efficacy of anti-stalkerware apps. Also, we found that the anti-stalkerware's response to stalkerware (user experience, information, and agency) affects users' trust. Hence, it would make sense to encourage and improve this kind of evaluation behavior. We suggest offering a toolkit for users to install on their phones. This toolkit should be able to install (and remove) a wide variety of stalkerware and dual-use software and track the anti-stalkerware's response. Such a toolkit would affect users in three ways: (1) all users would have the ability to safely and soundly evaluate their chosen tool's detection mechanism, (2) users could safely experience their tools response to malicious software, and (3) it would reduce the need to trust third-party reviews of anti-stalkerware apps. Similar to this approach, Parson et al. [66] suggest that a government body should track and evaluate anti-virus engines and publish public reviews. However, in contrast to our suggestion, users would then not experience their chosen app's response to threats.

Provide context-specific advice and give users agency. Detection ability is an important but not the only factor for users' safety. The type and amount of information apps present to users influence their response. Additionally, users' agency to respond to detected threats is crucial. Both information and agency need to be context-sensitive to the users' circumstances and the specific detected threats. For example, for IPS survivors safe responses to detected surveillance threats may be different before and after they have left their partner. This could include additional context-specific response options, e.g., generating fake location data or partially removing permissions without alerting the stalker. Without context-sensitive advice and user options, even an anti-stalkerware app with great detection ability may endanger users.

Leverage operating system's power to limit abuse. Improving anti-stalkerware apps and users' protection abilities is an individualistic approach to combating IPS. However, a systemic approach may be more effective in reducing IPS. Considering potential abuse in the design stage for operating systems, apps, and accessories may help fight IPS on a system level. Defensive design is a widely adopted approach across many disciplines. However, it focuses on unintentional errors in programming code and resulting apps. Other general approaches take intentional abuse into account at every step of the design process to mitigate interpersonal harm [68, 94]. Levy and Schneier [50] offered design considerations to ameliorate intimate privacy risks. Slupska and Tanczer [81] suggested an approach to threat model intimate partner violence in the design process. Interestingly, the two most common smartphone platforms, iOS and Android, are not equally susceptible to stalkerware targeted at consumer audiences [42, 66]. Parsons et al. report on the stalkerware industry [66] and the limited options to install these stalkerware apps on iOS without jailbreaking. Consequently, most commercial stalkerware for iOS devices rely on the target's iCloud account. Reputable companies do not want to publicly support dedicated stalkerware, so these apps are not published in app stores—or are quickly removed. This may result in a proliferation of other abuse-enabling dual-use apps (such as parental control apps) and their legitimate use-cases make them harder to police. Since legitimate use-cases are here to stay, it is necessary to adapt the design of these apps and the operating systems to limit misuse. The authors report recommendations applicable to platform providers that may curb stalkerware. They call for prominent, ongoing, and meaningful consent notices. These make it harder to install stalkerware surreptitiously on others' smartphones. Additionally, they call for on-device platform heuristics that detect misuse of ostensible dual-use software. Platforms have the power to disable abuse-enabling apps entirely – which may protect users unable to manage apps on their device. Platform providers have significant power over the kind of software they allow to run and which kind of app activities they make visible to users. Using this power would be an effective measure against the current stalkerware ecosystem.

8 Conclusion

Choosing effective anti-stalkerware solutions is a struggle. This case study evaluated two anti-stalkerware apps from multiple perspectives to understand users' selection and trust strategies. We identified five approaches that users apply: two based on user interaction, two based on the assumed detection abilities, and one on trusted third parties. All approaches are intuitive to apply and have some degree of legitimacy. However, the cognitive walkthroughs and reverse engineering approaches revealed severe drawbacks. We found that users' strategies do not inform them sufficiently about these apps and their abilities to mitigate violence, abuse, and harassment.

Our work helps improve current anti-stalkerware by suggesting design directions that increase users' trust and safety. These design directions focus on reassuring user experience, context-sensitive advice, and risk-appropriate user agency. Also, we suggest a user-deployable, toolkit-supported approach to evaluate anti-stalkerware's detection abilities and user experience. Such a toolkit-based approach builds on and encourages existing user behavior while improving its efficacy and safety. Lastly, while our study focuses on individualistic responses to anti-stalkerware, we emphasize the need for a systemic, platform-level approach to effectively combat intimate partner surveillance.

Acknowledgements

We thank the reviewers for their feedback on improving our paper. In particular, we thank our anonymous shepherd for their responsive, helpful, and kind guidance. The first author

conducted their work as part of the Saarbrücken Graduate School of Computer Science, Saarland University.

This research has received funding from the Vienna Science and Technology Fund (WWTF) through project ICT19-056, as well as SBA Research. SBA Research (SBA-K1) is a COMET Centre within the framework of COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMDW, and the federal state of Vienna. The COMET Programme is managed by FFG.

References

[1] Fiddler | Web Debugging Proxy and Troubleshooting Solutions. https://www.telerik.com/fiddler. (Accessed on June 8th, 2022).

[2] Frida • A world-class dynamic instrumentation framework. https://frida.re. (Accessed on June 8th, 2022).

[3] Jadx. https://github.com/skylot/jadx. (Accessed on June 8th, 2022).

[4] Alessandro Acquisti, Idris Adjerid, Rebecca Balebako, Laura Brandimarte, Lorrie Faith Cranor, Saranga Komanduri, Pedro Giovanni Leon, Norman Sadeh, Florian Schaub, Manya Sleeper, Yang Wang, and Shomir Wilson. Nudges for Privacy and Security: Understanding and Assisting Users' Choices Online. *ACM Computing Surveys*, 50(3):1–41, October 2017.

[5] Prerna Agrawal and Bhushan Trivedi. A Survey on Android Malware and their Detection Techniques. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–6, Coimbatore, India, 2019. IEEE.

[6] AppBrain. Anti spy mobile PRO: Changelog. https://www.appbrain.com/app/anti-spy-mobile-pro/com.antispycell, 2022. (Accessed on June 8th, 2022).

[7] Apple. App security overview. https://support.apple.com/guide/security/app-security-overview-sec35dd877d0/web, 2021. (Accessed on June 8th, 2022).

[8] Apple. App store review guidelines. https://developer.apple.com/app-store/review/guidelines/, 2021. (Accessed on June 8th, 2022).

[9] Daniel Arp, Michael Spreitzenbarth, Malte Hübner, Hugo Gascon, and Konrad Rieck. Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. In *Proceedings of the Network and Distributed System Security Symposium (NDSS),*, San Diego, CA, USA, 2014. Internet Society.

[10] AV-Comparatives. Android Test 2019 - 250 Apps. https://www.av-comparatives.org/tests/android-test-2019-250-apps/, January 2019. (Accessed on June 8th, 2022).

[11] Alessandro Bacci, Alberto Bartoli, Fabio Martinelli, Eric Medvet, Francesco Mercaldo, and Corrado Aaron Visaggio. Impact of Code Obfuscation on Android Malware Detection based on Static and Dynamic Analysis. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP*, pages 379–385, Funchal, Madeira, Portugal, 2018. SciTePress.

[12] Elsa Bakiu and Emitza Guzman. Which Feature is Unusable? Detecting Usability and User Experience Issues from User Reviews. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 182–187, Lisbon, Portugal, 2017. IEEE.

[13] Rosanna Bellini, Emily Tseng, Nora McDonald, Rachel Greenstadt, Damon McCoy, Thomas Ristenpart, and Nicola Dell. So-Called Privacy Breeds Evil": Narrative Justifications for Intimate Partner Surveillance in Online Forums. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3), December 2020.

[14] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006.

[15] Rahul Chatterjee, Periwinkle Doerfler, Hadas Orgad, Sam Havron, Jackeline Palmer, Diana Freed, Karen Levy, Nicola Dell, Damon McCoy, and Thomas Ristenpart. The Spyware Used in Intimate Partner Violence. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 441–458, San Francisco, CA, USA, May 2018. IEEE.

[16] Ning Chen, Jialiu Lin, Steven C. H. Hoi, Xiaokui Xiao, and Boshen Zhang. AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, pages 767–778, Hyderabad, India, 2014. ACM.

[17] Jerry Cheng, Starsky H.Y. Wong, Hao Yang, and Songwu Lu. SmartSiren: Virus Detection and Alert for Smartphones. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pages 258–271, San Juan, Puerto Rico, 2007. ACM.

[18] Sauvik Das, Adam D.I. Kramer, Laura A. Dabbish, and Jason I. Hong. The Role of Social Influence in Security Feature Adoption. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, pages 1416–1426, Vancouver, BC, Canada, February 2015. ACM.

[19] Alexander De Luca, Sauvik Das, Martin Ortlieb, Iulia Ion, and Ben Laurie. Expert and Non-Expert Attitudes towards (Secure) Instant Messaging. In *Proceedings of the Twelfth USENIX Conference on Usable Privacy and Security*, SOUPS '16, pages 147–157, Denver, CO, USA, 2016. USENIX Association.

[20] Verena Distler, Gabriele Lenzini, Carine Lallemand, and Vincent Koenig. The Framework of Security-Enhancing Friction: How UX Can Help Users Behave More Securely. In *New Security Paradigms Workshop 2020*, NSPW '20, pages 45–58, Online, USA, October 2020. ACM.

[21] Verena Distler, Marie-Laure Zollinger, Carine Lallemand, Peter B. Roenne, Peter Y. A. Ryan, and Vincent Koenig. Security - Visible, Yet Unseen? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2019)*, CHI '19, pages 1–13, Glasgow, Scotland, UK, 2019. ACM.

[22] Paul Ducklin. The google play "Super antivirus" that's not so super at all... https://nakedsecurity.sophos.com/2018/01/19/the-google-play-super-antivirus-thats-not-so-super-at-all-report/, January 2018. (Accessed on June 8th, 2022).

[23] Martin Emms, Budi Arief, and Aad van Moorsel. Electronic Footprints in the Sand: Technologies for Assisting Domestic Violence Survivors. In Bart Preneel and Demosthenes Ikonomou, editors, *Privacy Technologies and Policy*, pages 203–214, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[24] Álvaro Feal, Paolo Calciati, Narseo Vallina-Rodriguez, Carmela Troncoso, and Alessandra Gorla. Angel or Devil? A Privacy Study of Mobile Parental Control Apps. In *Proceedings on Privacy Enhancing Technologies*, volume 2020, pages 314–335, April 2020.

[25] Raffaele Filieri. What makes online reviews helpful? A diagnosticity-adoption framework to explain informational and normative influences in e-WOM. *Journal of Business Research*, 68(6):1261—1270, 2015.

[26] Diana Freed, Sam Havron, Emily Tseng, Andrea Gallardo, Rahul Chatterjee, Thomas Ristenpart, and Nicola Dell. "Is My Phone Hacked?" Analyzing Clinical Computer Security Interventions with Survivors of Intimate Partner Violence. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):202:1–202:24, 2019.

[27] Diana Freed, Jackeline Palmer, Diana Minchala, Karen Levy, Thomas Ristenpart, and Nicola Dell. "A Stalker's Paradise": How Intimate Partner Abusers Exploit Technology. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–13, Montreal, QC, Canada, 2018. ACM.

[28] Alisa Frik, Nathan Malkin, Marian Harbach, Eyal Peer, and Serge Egelman. A Promise Is A Promise: The Effect of Commitment Devices on Computer Security Intentions. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–12, Glasgow, Scotland, UK, 2019. ACM.

[29] FTC. FTC bans SpyFone and CEO from surveillance business and orders company to delete all secretly stolen data. https://www.ftc.gov/news-events/press-releases/2021/09/ftc-bans-spyfone-and-ceo-from-surveillance-business, 2021. (Accessed on June 8th, 2022).

[30] Bin Fu, Jialiu Lin, Lei Li, Christos Faloutsos, Jason Hong, and Norman Sadeh. Why People Hate Your App: Making Sense of User Feedback in a Mobile App Store. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1276–1284, Chicago, IL, USA, 2013. ACM.

[31] Asit Kumar Gahalaut and Padmavati Khandnor. Reverse engineering: An essence for software re-engineering and program analysis. *International Journal of Engineering Science and Technology*, 2(06):2296—2303, 2010.

[32] Arup Kumar Ghosh, Karla Badillo-Urquiola, Shion Guha, Joseph J. LaViola Jr, and Pamela J. Wisniewski. Safety vs. Surveillance: What Children Have to Say about Mobile Apps for Parental Control. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–14, Montreal, QC, Canada, April 2018. ACM.

[33] Arup Kumar Ghosh and Pamela Wisniewski. Understanding User Reviews of Adolescent Mobile Safety Apps: A Thematic Analysis. In *Proceedings of the 19th International Conference on Supporting Group Work*, GROUP '16, pages 417–420, Sanibel Island, FL, USA, 2016. ACM.

[34] Google. Android developers: Configure your build. https://developer.android.com/studio/build. (Accessed on June 8th, 2022).

[35] Google. Android developers: <manifest>. https://developer.android.com/guide/topics/manifest/manifest-element.html#package. (Accessed on June 8th, 2022).

[36] Google. Android developers: PackageManager. https://developer.android.com/reference/android/content/pm/PackageManager. (Accessed on June 8th, 2022).

[37] Google. Developer Program Policy: September 16, 2020 announcement - Play Console Help. https://support.google.com/googleplay/android-developer/answer/10065487, September 2020. (Accessed on June 8th, 2022).

[38] Xiaodong Gu and Sunghun Kim. "What Parts of Your Apps are Loved by Users?" (T). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 760–770, Lincoln, NE, USA, November 2015. IEEE.

[39] Emitza Guzman and Walid Maalej. How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 153–162, Karlskrona, Sweden, 2014. IEEE.

[40] Elizabeth Ha and David Wagner. Do Android users write about electric sheep? Examining consumer reviews in Google Play. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 149–157, Las Vegas, NV, USA, 2013. IEEE.

[41] Mahmoud Hammad, Joshua Garcia, and Sam Malek. A Large-Scale Empirical Study on the Effects of Code Obfuscations on Android Apps and Anti-Malware Products. In *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, pages 421–431, Gothenburg, Sweden, 2018. ACM.

[42] Diarmaid Harkin and Adám Molnár. Operating-System Design and Its Implications for Victims of Family Violence: The Comparative Threat of Smart Phone Spyware for Android Versus iPhone Users. *Violence Against Women*, 27(6-7):851–875, May 2021.

[43] Sam Havron, Diana Freed, Rahul Chatterjee, Damon McCoy, Nicola Dell, and Thomas Ristenpart. Clinical Computer Security for Victims of Intimate Partner Violence. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC '19, pages 105–122, Santa Clara, CA, USA, 2019. USENIX Association.

[44] Steffen Hedegaard and Jakob Grue Simonsen. Extracting Usability and User Experience Information from Online User Reviews. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2089–2098, Paris, France, 2013. ACM.

[45] Alex Hern. 'Fake' Android antivirus app developer says Virus Shield was a 'foolish mistake'. http://www.theguardian.com/technology/2014/apr/10/fake-android-antivirus-app-developer-virus-shield, April 2014. (Accessed on June 8th, 2022).

[46] Claudia Iacob and Rachel Harrison. Retrieving and analyzing mobile apps feature requests from online reviews. In *2013 10th Working Conference on Mining Software Repositories (MSR)*, pages 41–44, San Francisco, CA, USA, 2013. IEEE.

[47] Hammad Khalid. On identifying user complaints of iOS apps. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 1474–1476, San Francisco, CA, USA, 2013. IEEE.

[48] Younghwa Lee and Kenneth A Kozar. An empirical investigation of anti-spyware software adoption: A multitheoretical perspective. *Information & Management*, 45(2):109—119, 2008.

[49] Roxanne Leitão. Technology-Facilitated Intimate Partner Abuse: A qualitative analysis of data from online domestic abuse forums. *Human–Computer Interaction*, 36(3):203–242, 2021.

[50] Karen Levy and Bruce Schneier. Privacy threats in intimate relationships. *Journal of Cybersecurity*, 6(1), May 2020.

[51] Martina Lindorfer, Matthias Neugschwandtner, and Christian Platzer. MARVIN: Efficient and Comprehensive Mobile App Classification through Static and Dynamic Analysis. In *2015 IEEE 39th Annual Computer Software and Applications Conference*, COMPSAC, pages 422–433, Taichung, Taiwan, 2015. IEEE.

[52] Martina Lindorfer, Matthias Neugschwandtner, Lukas Weichselbaum, Yanick Fratantonio, Victor van der Veen, and Christian Platzer. ANDRUBIS - 1,000,000 Apps Later: A View on Current Android Malware Behaviors. In *2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, pages 3–17, Wroclaw, Poland, 2014. IEEE.

[53] Walid Maalej and Hadeer Nabil. Bug report, feature request, or simply praise? On automatically classifying app reviews. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, pages 116–125, Ottawa, ON, Canada, August 2015. IEEE.

[54] Diogo Marques, Ildar Muslukhov, Tiago Guerreiro, Konstantin Beznosov, and Luís Carriço. Snooping on Mobile Phones: Prevalence and Trends. In *Proceedings of the Twelfth USENIX Conference on Usable Privacy and Security*, SOUPS '16, pages 159–174, Denver, CO, USA, 2016. USENIX Association.

[55] Niels Raabjerg Mathiasen and Susanne Bødker. Threats or Threads: From Usable Security to Secure Experience?

In *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*, NordiCHI '08, pages 283—289, Lund, Sweden, 2008. ACM.

[56] Niels Raabjerg Mathiasen and Susanne Bødker. Experiencing Security in Interaction Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2325—2334, Vancouver, BC, Canada, 2011. ACM.

[57] Tara Matthews, Kathleen O'Leary, Anna Turner, Manya Sleeper, Jill Palzkill Woelfer, Martin Shelton, Cori Manthorne, Elizabeth F. Churchill, and Sunny 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*, CHI '17, pages 2189–2201, Denver, CO, USA, 2017. ACM.

[58] Stuart McIlroy, Nasir Ali, Hammad Khalid, and Ahmed E. Hassan. Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. *Empirical Software Engineering*, 21(3):1067–1106, June 2016.

[59] Andreas Möller, Stefan Diewald, Luis Roalter, Florian Michahelles, and Matthias Kranz. Update Behavior in App Markets and Security Implications: A Case Study in Google Play. In *Research in the LARGE: Proceedings of the 3rd International Workshop. Held in Conjunction with Mobile HCI*, pages 3–6, 2012.

[60] Ildar Muslukhov, Yazan Boshmaf, Cynthia Kuo, Jonathan Lester, and Konstantin Beznosov. Know Your Enemy: The Risk of Unauthorized Access in Smartphones by Insiders. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 271–280, Munich, Germany, 2013. ACM.

[61] Duc Cuong Nguyen, Erik Derr, Michael Backes, and Sven Bugiel. Short Text, Large Effect: Measuring the Impact of User Reviews on Android App Security & Privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 555–569, San Francisco, CA, USA, 2019. IEEE.

[62] OWASP. Mobile security testing guide (MSTG). https://mobile-security.gitbook.io/, 2021. (Accessed on June 8th, 2022).

[63] Danny Palmer. Can you trust your Android antivirus software? Malicious fake protection apps flood Google Play Store. https://www.zdnet.com/article/can-you-trust-your-mobile-antivirus-software-malicious-fake-protection-apps-flood-google-play-store/, June 2017. (Accessed on June 8th, 2022).

[64] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A. Visaggio, Gerardo Canfora, and Harald C. Gall. How can i improve my app? Classifying user reviews for software maintenance and evolution. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 281–290, Bremen, Germany, September 2015. IEEE.

[65] Sunoo Park and Kendra Albert. A Researcher's Guide to Some Legal Risks of Security Research. https://clinic.cyber.harvard.edu/files/2020/10/Security_Researchers_Guide-2.pdf, 2020. (Accessed on June 8th, 2022).

[66] Christopher Parsons, Adam Molnar, Jakub Dalek, Miles Kenyon, Bennett Haselton, Cynthia Khoo, and Ronald Deibert. The Predator in Your Pocket: A Multidisciplinary Assessment of the Stalkerware Application Industry. https://citizenlab.ca/docs/stalkerware-holistic.pdf, 2019. (Accessed on June 8th, 2022).

[67] Sameer Patil, Roberto Hoyle, Roman Schlegel, Apu Kapadia, and Adam J. Lee. Interrupt Now or Inform Later? Comparing Immediate and Delayed Privacy Feedback. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1415–1418, Seoul, Republic of Korea, April 2015. ACM.

[68] Eva PenzeyMoog. *Design for Safety*. A Book Apart, August 2021.

[69] Minh Vu Phong, Tam The Nguyen, Hung Viet Pham, and Tung Thanh Nguyen. Mining User Opinions in Mobile App Reviews: A Keyword-Based Approach (T). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 749–759, Lincoln, NE, USA, 2015. IEEE.

[70] Georgios Portokalidis, Philip Homburg, Kostas Anagnostakis, and Herbert Bos. Paranoid Android: Versatile Protection for Smartphones. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, Austin, TX, USA, 2010. ACM.

[71] Mila Dalla Preda and Federico Maggi. Testing android malware detectors against code obfuscation: A systematization of knowledge and unified methodology. *Journal of Computer Virology and Hacking Techniques*, 13(3):209–232, August 2017.

[72] Emilee Rader and Rick Wash. Identifying patterns in informal sources of security information. *Journal of Cybersecurity*, 1(1):121–144, September 2015.

[73] Emilee Rader, Rick Wash, and Brandon Brooks. Stories as Informal Lessons about Security. In *Proceedings of*

the Eighth Symposium on Usable Privacy and Security, SOUPS '12, Washington D.C., USA, 2012. ACM.

[74] Mizanur Rahman, Nestor Hernandez, Ruben Recabarren, Syed Ishtiaque Ahmed, and Bogdan Carbunar. The Art and Craft of Fraudulent App Promotion in Google Play. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 2437–2454, London, United Kingdom, 2019. ACM.

[75] Fahimeh Raja, Kirstie Hawkey, Pooya Jaferian, Konstantin Beznosov, and Kellogg S. Booth. It's Too Complicated, so i Turned It off! Expectations, Perceptions, and Misconceptions of Personal Firewalls. In *Proceedings of the 3rd ACM Workshop on Assurable and Usable Security Configuration*, SafeConfig '10, pages 53—62, Chicago, IL, USA, 2010. ACM.

[76] Audrey Randall, Enze Liu, Gautam Akiwate, Ramakrishna Padmanabhan, Geoffrey M. Voelker, Stefan Savage, and Aaron Schulman. Trufflehunter: Cache Snooping Rare Domains at Large Public DNS Resolvers. In *Proceedings of the ACM Internet Measurement Conference*, IMC '20, pages 50–64, Virtual Event, USA, 2020. ACM.

[77] Vaibhav Rastogi, Yan Chen, and Xuxian Jiang. DroidChameleon: Evaluating Android Anti-Malware against Transformation Attacks. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 329–334, Hangzhou, China, 2013. ACM.

[78] Elissa M Redmiles, Sean Kross, and Michelle L Mazurek. How I Learned to be Secure: A Census-Representative Survey of Security Advice Sources and Behavior. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 666–677, Vienna, Austria, 2016. ACM.

[79] Elissa M. Redmiles, Amelia R. Malone, and Michelle L. Mazurek. I Think They're Trying to Tell Me Something: Advice Sources and Selection for Digital Security. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 272–288, San Jose, CA, USA, 2016. IEEE.

[80] Kevin A. Roundy, Paula Barmaimon Mendelberg, Nicola Dell, Damon McCoy, Daniel Nissani, Thomas Ristenpart, and Acar Tamersoy. The Many Kinds of Creepware Used for Interpersonal Attacks. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 626–643, San Francisco, CA, USA, 2020. IEEE.

[81] Julia Slupska and Leonie Maria Tanczer. Threat Modeling Intimate Partner Violence: Tech Abuse as a Cybersecurity Challenge in the Internet of Things. In Jane Bailey, Asher Flynn, and Nicola Henry, editors, *The Emerald International Handbook of Technology-Facilitated Violence and Abuse*, pages 663–688. Emerald Publishing Limited, June 2021.

[82] Rick Spencer. The Streamlined Cognitive Walkthrough Method, Working around Social Constraints Encountered in a Software Development Company. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 353–359, The Hague, The Netherlands, 2000. ACM.

[83] Eric Spero and Robert Biddle. Out of Sight, Out of Mind: UI Design and the Inhibition of Mental Models of Security. In *New Security Paradigms Workshop 2020*, NSPW '20, pages 127–143, Online, USA, 2020. ACM.

[84] Coalation Against Stalkerware. The State of Stalkerware in 2019. https://media.kasperskycontenthub.com/wp-content/uploads/sites/100/2020/03/18084439/Kaspersky_The-State-of-Stalkerware-in-2019_Updated.pdf, April 2020. (Accessed on June 8th, 2022).

[85] Google Play Store. Anti spy mobile PRO. https://play.google.com/store/apps/details?id=com.antispycell, 2021. (Accessed on June 8th, 2022).

[86] Google Play Store. Mobile security - lookout. https://play.google.com/store/apps/details?id=com.lookout, 2021. (Accessed on June 8th, 2022).

[87] Yuan Tian, Bin Liu, Weisi Dai, Blase Ur, Patrick Tague, and Lorrie Faith Cranor. Supporting Privacy-Conscious App Update Decisions with User Reviews. In *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '15, pages 51—61, Denver, CO, USA, 2015. ACM.

[88] Emily Tseng, Rosanna Bellini, Nora McDonald, Matan Danos, Rachel Greenstadt, Damon McCoy, Nicola Dell, and Thomas Ristenpart. The Tools and Tactics Used in Intimate Partner Surveillance: An Analysis of Online Infidelity Forums. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1893–1909. USENIX Association, 2020.

[89] Kami E. Vaniea, Emilee Rader, and Rick Wash. Betrayed by Updates: How Negative Experiences Affect Future Security. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2671—2674, Toronto, ON, Canada, 2014. ACM.

[90] Arne Vidstrom. The legal boundaries of reverse engineering in the EU. https://vidstromlabs.com/blog/the-legal-boundaries-of-reverse-engineering-in-the-eu/, May 2019. (Accessed on June 8th, 2022).

[91] Artemij Voskobojnikov, Oliver Wiese, Masoud Mehrabi Koushki, Volker Roth, and Konstantin Beznosov. The U in Crypto Stands for Usable: An Empirical Study of User Experience with Mobile Cryptocurrency Wallets. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama, Japan, 2021. ACM.

[92] Rick Wash. Folk Models of Home Computer Security. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS '10, pages 1–16, Redmond, WA, USA, 2010. ACM.

[93] Cathleen Wharton, John Rieman, Clayton Lewis, and Peter Polson. The Cognitive Walkthrough Method: A Practitioner's Guide. In *Usability Inspection Methods*, pages 105–140. John Wiley & Sons, Inc., 1994.

[94] Karl Wiegers. Designing around bad actors and dangerous actions. https://uxdesign.cc/designing-around-bad-actors-and-dangerous-actions-8fc7984c510d, February 2021. (Accessed on June 8th, 2022).

[95] Zhen Xie and Sencun Zhu. AppWatcher: Unveiling the Underground Market of Trading Mobile App Reviews. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '15, pages 1–11, New York, NY, USA, 2015. ACM.

[96] Zhen Xie, Sencun Zhu, Qing Li, and Wenjing Wang. You Can Promote, but You Can't Hide: Large-Scale Abused App Detection in Mobile App Stores. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, ACSAC '16, pages 374–385, Los Angeles, CA, USA, 2016. ACM.

[97] Jinjian Zhai, Humayun Ajmal, and Jimmy Su. Preying on Insecurity: Placebo Applications With No Functionality on Google Play and Amazon.com. https://www.fireeye.com/blog/threat-research/2014/06/preying-on-insecurity-placebo-applications-with-no-functionality-on-google-play-and-amazon-com.html, June 2014. (Accessed on June 8th, 2022).

[98] Min Zheng, Patrick P. C. Lee, and John C. S. Lui. ADAM: An automatic and extensible platform to stress test android anti-virus systems. In Ulrich Flegel, Evangelos Markatos, and William Robertson, editors, *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 82–101, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[99] Yixin Zou, Kevin Roundy, Acar Tamersoy, Saurabh Shintre, Johann Roturier, and Florian Schaub. Examining the Adoption and Abandonment of Security, Privacy, and Identity Theft Protection Practices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–15, Honolulu, HI, USA, 2020. ACM.

A Codebook for the Thematic Analysis

Table 1 shows the initial codebook. Table 2 shows the codebook we used to focus on the users' perception of the case-study apps' safety and security.

Table 1: Initial codebook that included users' general perceptions about the apps.

CODES	DESCRIPTION	ANTISPY	LOOKOUT	TOTAL
Effect +	Review reports an event that demonstrated the app's efficacy	119 110	41 39	160 149
Experience +	Review focuses on the app's great user experience	155 161	19 12	174 173
Performance +	Review highlights the technical performance of the app (e.g., quick scans or low battery drain)	40 37	12 14	52 51
Usability +	Review reports that the app is easy to understand and/or use	20 20	11 10	31 30
Payment +	Positive experience with payment for the app itself or the subscription	18 19	3 3	21 22
Response +	Positive experience with responsive app developers or support team	8 7	10 12	18 19
Privacy +	Reviewer praises the app for its privacy-preserving approach	5 4	4 5	9 9
Effect -	Review reports an event that demonstrated the app's inadequacy	28 27	27 23	55 50
Experience -	Review focuses on the app's bad user experience	1 2	1 0	2 2
Performance -	Review highlights the bad technical performance of the app (e.g., battery drain, slow scans, or bugs)	94 101	18 18	112 119
Usability -	Review reports that the app is hard to understand and/or use	47 42	15 18	62 60
Payment -	Negative experience with payment for the app itself or the subscription	48 44	20 21	68 65
Response -	Negative experience with unresponsive app developers or support team	22 21	2 2	24 23
Privacy -	Reviewer perceives the app as privacy-infringing	5 4	4 3	9 7

Table 2: The codebook for the second coding iteration that focused on the users' perception of the app's effectiveness, i.e. the *effect* code in the previous codebook.

CODES	DESCRIPTION	ANTISPY	LOOKOUT	TOTAL
Real Life Safe	Experience report of an event where app protected reviewer from harm	52 50	27 26	79 76
Test passed	Reviewer tested the app's detection capabilities and was satisfied by the results	5 6	3 3	8 13
Secure Feeling	Experience of using the app gave reviewer a feeling of security	67 70	9 9	76 79
Notifications	Prompt notifications about security incidents gave reviewers a secure feeling	17 19	2 2	19 21
Real Life Fail	Experience report of an event where app failed to protect reviewer from harm	13 12	8 8	21 20
Test Fail	Reviewer tested the app's detection capabilities and was not satisfied by the results	13 15	8 9	21 24
Insecure Feeling	Experience of using the app did not reassure reviewer about its security	8 9	10 10	18 19
Likes Feature	Reviewer praise a specific feature of the app	10 10	3 3	13 13
Misses Feature	Reviewer complains about a feature they had before or would like to have	27 24	5 5	32 29
Update	Review concerned changes to the app by a software update	13 14	2 2	15 16
Time of Experience	Reviewers reference their long usage experience with the app to communicate their trust in the app's capabilities	19 19	3 4	22 23