Creating Transparency to Raise User's Awareness on Data Collection of Mobile Apps: A Literature Review

Chen Yang
chen.yang@stud.tu-darmstadt.de

Shabnam Sohrabi
shabnam.sohrabi@stud.tu-darmstadt.de

Abstract

The lack of privacy awareness, specifically the discrepancy between stated privacy policies and actual data collection in smartphone ecosystems prevents users from being able to compare apps in terms of privacy-friendliness, from making informed privacy decisions and eventually protect them self from the disclosure of personal data. In this paper we analyzed how the discrepancy between privacy policies and actual app's behavior can be identified as well as analyzed a chosen set of existing Tools to raise user's awareness of privacy-invasive app behavior. The tools consists of ex-ante privacy enhancing tools as well as ex-post privacy enhancing tools. The tools are getting examined in regards to their capabilities, methods and limitations. To examine the tools we conducted a literature review. We collected and contrasted recent tools and possible solutions. The results showed that the tools indeed are on the right path to empower users to make better decisions in regards to the choice of apps. Further, we found that due to the limitations these methods and tools leave room to improvement, and hence additional investigation is needed.

Keywords: smartphone ecosystems; android; privacy; privacy policies; permission; privacy concern; privacy behavior; data collection; extraction

1 Introduction

Mobile apps are well-integrated in our daily lives. They provide us with all kinds of functionalities and services and seem to make our lives easier. While most apps seem to be free of any costs, we do share a lot of our personal information. Mobile apps can collect sensitive data and use it for monetary purposes (e.g. by sharing data to third-parties). In the process users' privacy can get violated. Meanwhile the user's have very little awareness about what they share, what happens to their data and what threats they get exposed through it. The general lack of transparency and lack of privacy awareness in mobile ecosystems leaves users unable to make informed privacy decisions (Hatamian et al. (2018)). The question that then naturally arises is: *How to raise users' awareness by informing them about this data collection?*

In this study, we examine, compare and contrast existing tools, methods and solutions to identify privacy-invasive app behavior with a special focus on privacy policies as well as app permissions. By leveraging a literature review as a reference point, an overview on how to raise user's awareness by informing them about mobile data collection will be given. This study's aim is twofold: (1) identify how the discrepancy between personal data collection of mobile apps and privacy policies can be discovered, (2) give an overview of existing privacy-enhancing tools mainly based on app permissions for smartphone users to support them for informed privacy decision-making and raise their awareness. The rest of the paper is organized as follows: Section 2 reviews and compares the existing works in the literature related to detect contradictions from privacy policies. Section 3 describes and discusses the chosen set of privacy enhancing tools for smartphone users after having introduced related work in this field. Finally, we conclude the paper and point to the future directions of research in Section 4.

2 Identifying data collection through privacy policies

In this section we will investigate how the discrepancy between personal data collection of mobile apps and privacy policies can be discovered. First, we will present related work on this topic and dis-

cuss some tools that can discover the discrepancy and compare the differences and similarities of the tools. Second, we take a closer look at the paper on different tools, however, it reveals a number of gaps and shortcomings. Furthermore, more specific research questions will be introduced and investigated in these tools. Finally, we concluded with possible solutions in the future.

2.1 Related work

Previous studies showed that NLP is a good approach for analyzing privacy policies automatically and data mining or deep learning models are used to extract useful information from policies. Some other studies have used crowdsourced ontologies for analyzing policies. However, these approaches are restricted by correctness, incompleteness and complicated collection. The reason is that these approaches do not have the method to extract usable ontologies from such information and rely on a fixed lexicon. Some previous studies(Zimmeck et al. (2016). and Yu et al. (2016).) have also found negative statements in privacy policies. As mentioned in Andow et al. (2019)., these two tools used bi-grams to detect the negative statements. In comparison to these two previous work, PolicyLint is a privacy policy analysis tool and can offer a thorough analysis based on automatically constructed ontology and find negations and exceptions in text. Privacy policies always state that they do not collect or share personal user data while the data usage in app's true behavior fail to obey these rules. Many previous works show that there are contradictions in privacy policies. Bui et al. (2021). describe how they extract and verify data usage purposes, privacy statement and data flow. To evaluate their method, they present the results on 23,144 apps with a valid privacy policy. The results reveal that there are 29,521 potentially contradictory sentence pairs in 3,049 (18.14%) privacy policies. Andow et al. (2019). describe the design of PolicyLint and how it can identify contradictions by recognizing negation and varying semantic levels of data objects and entities. To evaluate their method, they present the results on 11,430 privacy policies from top Android apps. The results reveal that about 17.7% of the policies included logical contradictions and narrowing definitions, with 14.2% containing logical contradictions. In order to analyze the purpose of app behavior, Whyper Pandita et al. (2013)., AutoCog Qu et al. (2014). and CHABADA Gorla et al. (2014). figure out the inconsistencies between app's behavior and app's descriptions. Furthermore, after years of developing Natural Language Processing (NLP) and Machine Learning (ML) can provide better support for researchers to analyze privacy policies. Since privacy policies are long, vague and difficult for experts and algorithms to interpret. Privee Zimmeck and Bellovin (2014). and Polisis Harkous et al. (2018). analyze privacy policies at the document- and paragraph level to answer users' questions Andow et al. (2019). Therefore some research focuses on how to measure the usability and effectiveness of privacy policies. The result has shown that it is difficult for users to understand the privacy policies and the suggestion is that the privacy policies should be simplified. Cranor et al. (2016). 's work utilized a large-scale study of privacy notices of US financial institutions to highlight a number of related practices."Andow et al. (2019). Instead of using standardized models for notices, the privacy policies in mobile apps do not follow standardized models which is also a challenge.

2.2 Discussion

Methodology In Bui et al. (2021).'s work, there are three parts of the experiment – extraction of data usage purpose clauses, privacy statement extraction and data flow extraction. In the extraction of data usage purpose clauses phase, it is divided into extraction of data practice predicates and semantic arguments and extraction of purpose clauses. In the first process, PurPliance analyze patterns of semantic arguments, syntactic structures and a lexicon of data practices. Then it finds the verbs that have shown the intention of "collect" and "share", which they already have created a table and listed Sharing-Collection-or-Use(SCoU) verbs. In their previous work PolicyLint only distinguished between collection and sharing of data because they chose common words from randomly selected privacy sentences to extend Sharing-or-Collection(SoC) verbs. However, every verb can have a different meaning in sentences and some of them are irrelevant to the data collection/sharing/use. In order to get clearer usage of the verbs in different sentences from the privacy policies corpus, they added a verb in SCoU list to survey its usage. Furthermore, they selected verbs that are frequently used to express data practices so that both of the recall and precision can increase.

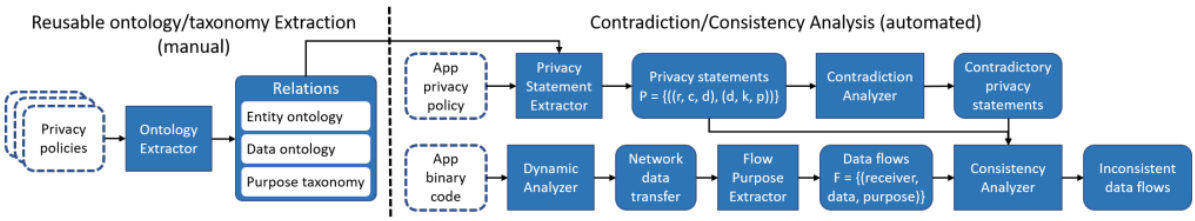

Figure 1: PurPliance system workflow. Dashed boxes indicate the system inputs.(Bui, Duc, et al.2021, Fig.1.)

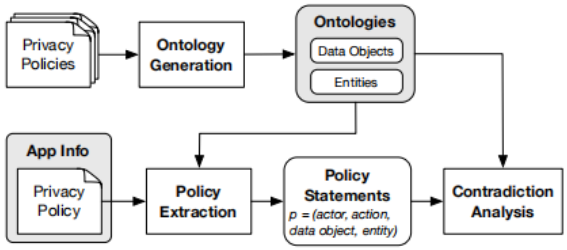

Figure 2: Overview of PolicyLint(Andow, Benjamin,et al.2019, Fig.1.)

Data Practice	Verbs
Sharing	disclose, distribute, exchange, give, lease, provide, rent, release, report, sell, send, share, trade, transfer, transmit
Collection	collect, gather, obtain, receive, record, store, solicit
Use	access, analyze, check, combine, connect, keep, know, process, save, use, utilize

Table 1: List of the SCoU verbs used by PurPliance.(Bui, Duc, et al.2021, Table.1.)

What they acquired from the data practice predicate is that the arguments of the same event can be expressed consistently by syntactic forms and parameters of privacy statements. From their example, the sentences below have the same data-usage event:

We do not share your personal data with third parties for target ads.

Third parties may not collect your personal data to deliver targeted ads.

In this example, the verbs are for and to, while the purpose of delivering targeted ads is the same. Therefore even though the syntactic structure of the sentence is different, the semantic arguments of the events are considered consistently.

Based on a specific Corpus–CoNLL2012 corpus' verb sense frames Weischedel et al. (2013). they identified some semantic arguments that represent purposes. The common arguments argument modifier purpose and purpose-not-cause are shown in Table 2.

Besides these two common purpose arguments, PurPliance also add arguments "use" and "save" to predicate the certain purposes. Due to the multiple meanings of a verb in different situations, they verified that the data practice verbs shown in context are all relevant for privacy purposes. Here's an example: the verb "save" has the meaning of save money or to collect (accurate) things. The latter meaning is more close to their case about data collection. However, they did not disambiguate the multiple meanings in their analysis. The reason is that the meaning of these data practice verbs are more relevant with the data usage purposes instead of other usages in privacy policies.

In classification of policy purposes process, it consists of uncompounded purpose extraction, purpose taxonomy and data-usage purpose classifier. Uncompounded purpose extraction means that PurPliance decomposes complicated purpose clauses into simple single-purpose parts and are represented by a predicate pair (PO). Here are examples of PO pairs. "to provide and improve our services" is decomposed into (provide, our service) and (improve, our services). Likewise, "for fraud prevention and service maintenance" can be converted to "fraud prevention" and "service maintenance". After extracting uncompounded purpose clauses from privacy policies, they classified them into semantically-similar groups by using text clustering Manning and Schutze (1999). They use a BERT-based sentence embedding model trained on semantic textual similarity data-sets to convert each clause into real-value vectors. They define Provide ad and Personalize ad as distinct classification. Providing ad means only deliver, show, or provide advertising while personalizing ad means specially customize advertising. Furthermore, they grouped these low-level purpose into high-level purposes(Production, Marketing, Legality and other), such as Provide ad is a Marketing

Data Action	Sender	Receiver	Data	Purpose	Example
Sharing	Arg0	Arg2	Arg1	Argm-Prp or Argm-Pnc	[We]$_{Arg0}$ do not [share]$_{V}$ [your data]$_{Arg1}$ [with third parties]$_{Arg2}$ [for their purposes]$_{Argm-Pnc}$.
Collection	Arg2	Arg0			[We]$_{Arg0}$ [collect]$_{V}$ [passwords]$_{Arg1}$ [for authentication]$_{Argm-Prp}$.
Use	N/A	Arg0			[We]$_{Arg0}$ may [process]$_{V}$ [your contact information]$_{Arg1}$ [to send you promotions]$_{Argm-Prp}$.

Table 2: List of the SCoU verbs used by PurPliance.(Bui, Duc, et al.2021, Table.2.)

purpose while Personalize ad is categorized under Provide and Personalize service. Next step PurPliance classifies purpose clauses by matching predicate patterns and object patterns and one clause might be categorized into multiple categories. In order to have good performance on classification, they extracted 198,339 purpose clauses from their privacy policy corpus of 16.8k unique privacy policies and divided it into 158,671(80%) training sets and 39,668(20%) test sets. Finally PurPliance achieved 97.8% precision on average.

In the privacy statement extraction phase, they first gave the definition of privacy statement. Each sentence in a privacy policy can be taken as a privacy statement with two components: Data Collection refers to transfer data to a receiver and Data Usage refers to the usage and purpose of the data. Then PurPliance extracted phrases in 3 steps: (1) identify data practice predicates, (2) extract the semantic arguments of each predicate and (3) map these arguments to the parameters.Bui et al. (2021).

In data flow extraction, according to the content of the data, the destination and the app description they can infer the data types and purposes of app-server communications. The function of PurPliance here is to distinguish first and third parties by analyzing the receiver and the inferred data-usage purpose. Compared to PoliCheck, PurPliance can get fine-grained intentions of data usage. Here's an example of what they have found: An app named Wego flights sent a Client ID to their own server to collect user ID for Marketing Analytics, which means the first party can collect data for its own purposes.

Here are five types of data see Table3.

In order to classify these types of data, there are 4 types of classifiers (Logistic Regression(LR), Multi-Layer Perceptron(MLP), Random Forest(RF) and Support Vector Machine(SVM)) to classifier fea tures, which Random Forest was found out with the best result 95% F1 score, 97% precision and 93% recall rate on average.

In the data traffic purpose inference process, they used the same features to predict the purpose of transferred data. These features are destination URL, sent data with the form of "//host/path", the characteristics of the data type and the relationship between the app and the server. Similar to the data type classification, they also used the above 4 classifiers. Their result showed that the transferred data is the most effective feature.

Bui et al. (2021). focus on analyzing policy contradictions and flow-to-policy inconsistencies. For example, they found out that there is a discrepancy between collected personal data and privacy policies. There are three main findings that show the discrepancy in different scenarios. First, although lots of apps state that they only use personal data for internal purposes,which contradicts another statement in private policy. Second, there are contradictory statements in common privacy policy templates. Third, the apps promise that they do not share user's personal data with third parties or for marketing purposes.However, according to another sentence in the policy, it indicates the possibility that they may collect information about the user's online activities.

Differences All the papers propose a tool for discovering the discrepancy between the personal data collection of mobile apps and privacy policies. These two tools have differences and similarities since PurPliance was published after PolicyLint and PolicyLint was the first tool to uncover contradictions. According to the table, PurPliance has better performance on metrics such as precision and recall rate because it uses larger dataset and more state-of-the-date NLP technology. More precisely, the results indicate that PurPliance enhances the precision from 19% to 95% and recall from 10% to 50% in comparison with PolicyLint. PurPliance used the same method in PolicyLint to analyze its dependency tree to determine its negation. PoliCheck is also a tool for analyzing privacy policies that developed on PolicyLint and has overcome the problem that previous work can not differentiate the entity such as first-party and third-party. However, the usage of PurPliance is wider than PoliCheck. For example, the researchers of PoliCheck do not focus on the business purpose

Data type	Example
identifiers	hard/software instance and advertising IDs
network information	types of network
device information	device types and configurations
location	GPS coordinates
user account information	user name, password and demographics

Table 3: Five types of data based on Bui, Duc, et al. 2021, Table 5.

of the data flows while PurPliance is general to all app categories and has no limitations on any type of apps. PPChecker is a tool that can automatically identify five kinds of problems in privacy policy. The five problems contains incomplete privacy policy, incorrect privacy police, imprecise privacy policy, inconsistent privacy policy and user-friendly privacy policy.

Based on the four articles, these different aspects are showed as Table 4.

Limitations There may be some possible limitations in these studies. PolicyLint is limited by the past NLP techniques such as the limitations of NLP parsers and named-entity recognition and can not extract the conditions of purposes behind collections and sharing statements as well. It also only focuses on the policies of Android apps rather than iOS or web. PoliCheck is built on top of PolicyLint thus it inherits the limitations too. In addition, PoliCheck only tracks some specific data types like name, location, phone number and so on. Furthermore, PoliCheck can differentiate the first-party and third-party. However, it has the potential of misclassifying these two. The limitation of PPChecker is the insufficient privacy policies and defective system design. PurPliance is also limited by the current Natural Language Processing research. For example,Semantic Role Labeling is still a challenging task.Because even the state-of-the-art SRL model still could be improved with only 87% F1 score and 85.5% recall rates. Additionally, due to the unspecifically trained privacy policy domain dataset, PurPliance has low performance. The second limitation concerns extracting the data flows from network traffic. PurPliance is unable to decode certificate-pinned traffic and exercise login-required apps that use external verification information. As for the analysis method, their model PurPliance only focuses on client-side information, which leads to the limited performance on observing the later purpose of processing on the servers. They assume that even the names of app resources such as package name or URL hosts/paths are meaningful, but they can not reveal the true intentions of data flows. Therefore, the extraction is not always reliable and they still need the anonymous aggregate information that is automatically collected from the apps to extract the purpose from server-side,since this information is lost once it is received by the servers. Furthermore, it also suggests human verifying the detections but only specialists such as regulators and service lawyers.

3 Identifying data collection through app permissions

In the last section we looked how the discrepancy between personal data collection of mobile apps and privacy policies can be discovered by tools from a technical point of view. In this section we will investigate how data collection can particularly be identified through app permissions and get eventually disclosed to the user. For this purpose, we will pay special attention to analyzing existing tools and methods in research.

We start by introducing crucial literature and findings for our research question. We then proceed with the presentation, analysis and comparison of the selected tools. We conclude with limitations and possible research questions in the future.

3.1 Related work

App permissions have to be given by the user so that mobile apps can gain access to the smartphone's resources (e.g. location, camera, contacts) to provide certain functionalities. Although since Android Version 6.0 (see table 5 for a comparison) users are able to restrict the requested permissions later on at run-time of the app, prior studies have shown that neither many users are aware of it nor make use of it. For example as Hatamian et al. (2019b). states it is still not effective because it is not comprehensible for the users and they tend to value apps more than their privacy at times. The phenomenon of the contradiction between users

	Andow et al. (2019).	Andow et al. (2020).	Yu et al. (2021)	Bui et al. (2021).
Tool	PolicyLint	PoliCheck	PPChecker	PurPliance
Source	11430 popular apps	13796Android apps	2500 popular apps	23,1k Android apps
Metric	97.3% precision (496/510) based on the 14 false positives identified	Achieve an overall 90.8% precision (139/153) for performing flow-to-policy consistency analysis.	Achieve the 95% precision and 50% recall rate	Achieve the 78.7% precision and 81.6% recall rate
Innovation point	Identify contradictions within individual privacy policies for software.	Differentiate the entity	Correlating UI Elements and Texts	Find the purpose of the apps' data collection
Conclusion	Around 17.7% of the policies contain logical contradictions and narrowing definitions, with 14.2% containing logical contradictions.	Up to 42.4% of applications either incorrectly disclose or omit disclosing their privacy-sensitive data flows.	1,850 (74.0 percent) apps' privacy policies having at least one problem.	PurPliance significantly outperforms a state-of-the-art method and detect contradictions/inconsistencies in a large number of Android apps.

Table 4: Comparision table

privacy attitudes and actual behavior is also known as the privacy paradox in research (Hatamian et al. (2018)).

Hindrances to making informed privacy decisions are grounded in information asymmetry and lack of transparency (Hatamian et al. (2018)). Users are not well-aware of the potential negative consequences of threats like tracking spyware, phishing, unintended data disclosure, targeted ads or spam when using privacy-invasive apps (for further Information on identified threats see (Hatamian et al. (2019a)). A wealth of previous work has examined users' perceptions and desires for smartphone privacy. Wijesekera et al. (2015) revealed through their study the discrepancy between users' expectations and actual app behavior. 80% of the Study's participants would have preferred to prevent at least one permission request. All in all they would have liked to block over a third of all requests. They confirmed the need of transparency in regard to what app accesses which resources and at what frequency. Another Study by Crager and Maiti (2017) shows users learning about the threats were immediately concerned about their privacy. While users tend to be concerned about privacy, they are not informed or empowered enough to protect themselves. It is challenging and difficult for the users to compare apps' privacy friendliness and performance and to protect their own privacy. One of the reasons are for example as stated in the previous section the discrepancy between privacy policies and the actual apps' behavior, the complex legal language among other things that make it difficult for users to prevent the disclosure of their personal data (Hatamian et al. (2021)). These reasons emphasize the importance to create transparency for the user's by providing quantifiability and thus comparability of apps in regard to their privacy impact (Hatamian et al. (2017)).

One possible solution to create more transparency is through ex-ante (pre download of the app) and / or ex-post transparency (post installment of the app) enhancing tools (Hatamian (2020)). In the following we will focus on a chosen set of transparency enhancing tools (see table 6 for an overview). Based on Hatamian et al. (2019a) we define privacy-friendliness: when apps request fewer number of dangerous permissions, have less discrepancy between manifest and available clarification in policy document, have reasonable permission usage during run-time and expose fewer threats.

< Android Version 6.0	≥ Android Version 6.0 (Release Oct. 2015)	IOS
• permissions had to be granted at install time by the users • not possible to restrict permissions later (Hatamian et al. 2018, p. 2) • No information about frequency, volume or amount of personal data retrieved and transferred (Hatamian et al. 2019, p. 89)	• more control to the users (revision /revoke permissions at run-time possible) (Hatamian et al. 2018, p. 2)	• access to resources is checked upon time-of-use (Amini et al., 2014)

Table 5: An overview of operating system's app Permissions manager

3.2 Discussion

We structure the chosen tools and the according Literature into 3 categories. The first category will focus on Hatamians work, which includes the Android App Behaviour Analyzer (Hatamian et al. (2018)) and his multiperspective approach (Hatamian et al. (2019a), Hatamian et al. (2021)). The second category will focus on TaintDroid and similar approaches. The last category will focus on improving the operating system's app manager.

First category Such a privacy enhancing tool was introduced by Hatamian et al. (2018) new tool called Android Apps behavior analyzer (A3). This tool analyzes installed apps for potential invasive activities on Android devices and notifies the user about violations. An example for this can be the violation of the Principle of Least Privilege (PoLP), hence when an app requests to access data that is not needed for it's functioning (over privileged) (Hatamian et al. (2018)). The capabilities of the A3 consists of specific app selection, individual scan intervals, enabled permission restriction, and the behaviour analysis of the app. The conducted user study highlighted the helpfulness of the A3 to users by collecting and contrasting privacy concerns and perceptions before and after using the tool. It shows that tools like the A3 do raise user's privacy awareness and enable informed privacy decision-making (Hatamian et al. (2018)). The technical implementation of the A3 (see figure 3) is as follows: the log reader component reads the device's logs and produces raw data. The data mining component then analyzes the raw data in regards to the apps' privacy behavior by a rule-based approach. For that they defined a set of sensitive permissions (for example the access on sensitive resources while the user does not seem to use it's device). The results in the form of detail of the access (date, time and reason together with a short explanation) are then sent to inform the user. Therefore, the users can transparently manage their resource accesses. The A3 works without any need for modification of the OS or root access and can be installed on recent Android versions (Hatamian et al. (2018)). Hatamian et al. (2019a) further developed a multilateral approach based on unlike before, a multi-source method for privacy analysis and data extraction transparency. They revealed that majority of the fitness apps (7 out of 10) kept accessing dangerous permissions, despite having no user interaction. This multilateral method is a security analysis approach that includes all stakeholders' perspectives and needs in a security analysis. Hatamian et al. (2019a) provides a more comprehensive measurement and uses a four-pillar methodology including Information from Permission Manifest Analysis (data access intentions, developers declare use of so-called sensitive permissions that grant access to data such as call logs, contact lists), privacy policy

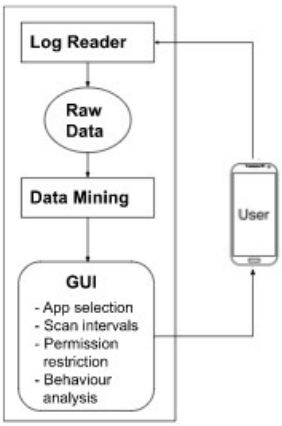

Figure 3: A high-level overview of the A3 based on Hatamian et al. 2018, Fig.1.

Source	Amini et al. 2013, 2014	Bar et al. 2015	Enck et al. 2014	Hatamian et al., 2018	Hatamian et al., 2019, 2021	Bock et al., 2021
Tool / Method	AppScanner, Gort	Styx	Kirin, TaintDroid	Android Apps Behaviour Analyser (A3)	A Multilateral Privacy Impact Analysis Method	Maybe-button developed app
Description, Aim	evaluate mobile apps privacy on a large scale, provides transparent & descriptive information about app behavior	privacy risk communication system, improves the comprehensibility and comparison of apps	identifies when apps use a dangerous combination of permissions and/or action strings	notifies about potential privacy invasive activities	Privacy-friendliness score, quantifies comparison e level of personal data usage of apps before installation and during installation	Partial consent other than the usual binary (allow/deny) options, time-bound partial consent
analyzed topic, outcome	privacy expectations, privacy sensitive app behavior legitimate to user or not	privacy risk information based on the second-order privacy risk perspective	protect against and detect suspicious apps	smartphone users' concern and expectation	Privacy-friendliness Risk Estimator	reassess trade-off between service and privacy consequences

Table 6: privacy enhancing tools

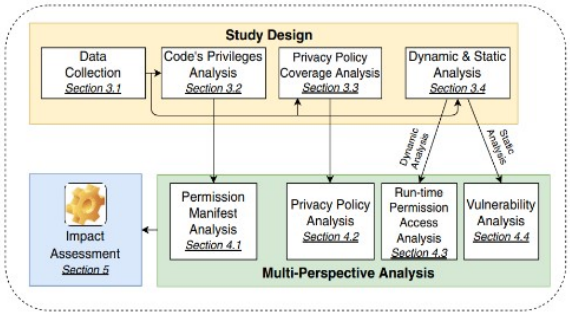

Fig. 1 A high-level overview of multi-perspective privacy and security analysis of COVID-19 contact tracing apps

Figure 4: A high-level overview of the multi-perspective-method (Hatamian et al. 2019, Fig.1.)

Analysis (what the developer's request (in manifest) and what they do (actual permission usage)), Permission Usage Analysis (as in Hatamian et al. (2018)) and User Reviews Analysis (extract comments on perceived app privacy problems, privacy threat classification. While Hatamian et al. (2018) A3 conducts an app behavior analysis, this study goes further and includes app vendors, end user feedback and actual app behavior. All this Information gets synthesized by a scoring algorithm and the outcome is a privacy-friendliness risk estimator. While all permission accesses are seen to be equally risky for privacy, the scoring (total of 36 points) works as follows: 10 for requesting permissions, 10 for not clarifying purposes in privacy policy, 10 for accessing permissions when the app is not in use and 6 for identified threats from user review analysis (Hatamian et al. (2019a)). This method was conducted in a case study on ten popular fitness apps. This approach was used again in Hatamian et al. (2021) to analyze the privacy and security performance of 28 contact tracing apps available on Android platform from various perspectives, including their code's privileges, promises made in their privacy policies, and static and dynamic performances. Through this method, they analyze the permissions declared in the Android manifest files. In particular, they focus on the dangerous permissions and analyze the privacy policies of these apps. They monitor app behavior by logging in each resource access event during run-time.

Second category Enck et al. (2014) aims to provide users with better visibility on how third-party apps collect and share their private data. For that the framework TaintDroid a system-wide dynamic taint tracking and analysis system was designed. It is able to simultaneously track multiple sources of sensitive data. TaintDroid enables real-time-analysis by using Android's virtualized execution environment. The work of Enck et al. (2014) has been very important. Many researchers have used Taintdroid as an extension to their own work. This is for example the case for Bar et al. (2019). They address the conceptualization of long-term privacy risks of mobile app usage. Therefore they designed Styx (privacy risk communication system)

for Android. In regards to the effectiveness of the risk communication, they found out in a proof-of-concept-study that Styx provides more comprehensible privacy-risk information while improving users' risk and trust perceptions. Simplifying the comparison of apps. Bar et al. (2019) classifies privacy risks into two groups: first-order privacy risk (single information leakage) and second-order privacy risk (threats through user-profiling and data-mining). Unlike before this tool focuses on the latter. Amini et al. (2013) introduces AppScanner that is in distinction from the other tools an automated cloud-service based on crowdsourcing, virtualization, and automation. It's objective is to to enable the large scale app behavior analysis by it. Appscanner is a prototype and consists of several sub modules and also uses the extension TaintDroid. The module App Mapper analyzes the app and results into a control flow graph of the main screens, associated privacy-related behaviors, and main functionalities. This will be then visualized interactively and presented to the users in another module. The remaining modules (Crowd-Scanner, Privacy Evaluator, Privacy Summarizer) still need to be developed. While AppScanner and TaintDroid have in common that they detect sensitive information leaks in mobile apps, AppScanner additionally attempts to present this information in a comprehensible way to the end-users.

Third category In the last category we will consider in passing an alternative approach than the previous ones: the partial consent or the "maybe-button" proposal for operating systems app permission by Bock et al. (2021). So far only a binary selection is available to the users when granting rights to use an application: namely accept and reject (see table 3). Through the "maybe-button" the users are given an additional option when an app permission request is made, to limit possible exploitation of personal data. It is a time-bound partial consent, which means that after a certain amount of time the user is given the chance to reassess the trade-off between the apps providing functionalities and privacy consequences. citetbock2021partial tested this through an developed app that requested diverse access permissions. The results showed that only one fifth of the participants used this option, while under half of the participants stated that they would like to use it on their private devices. Currently research on this suggestion seems to be still in its infancy and needs further investigation.

Comparison While all chosen methods use one form of an Ex-post transparency generating approach to analyze app behavior / permission usage. Hatamian et al. (2019a), Hatamian et al. (2021)) is the only one with a holistic approach. In table 7 the different used analysis methods are depicted. After realizing the A3 tool, he extended it by including many more analysis methods, like the permission manifest analysis, privacy policy analysis and user review analysis. Hence Hatamians work is the only one providing users transparency before and after installing the app. All of the chosen works focus on the Android platform, which has been the platform of choice for researchers. This likely results because: 1) Androids dominance in the market, hence a larger user base (Bock et al. (2021)) 2.) Android is open source, allowing modifications (rooting) and researchers to build prototypes to validate their ideas for real applications (Enck et al. (2014)) and 3) is generally believed to be exposed to higher risks. In general different kind of solutions categories exist. They can be divided into the app permission based solutions (focus of this section; main contributions by Hatamian), as well as network based solutions or a proposed improvements to the operating system's app permissions manager. It is very well possible that other approaches exist as well in research, but this would go beyond the scope of this seminar paper.

Limitation As we have seen the privacy-friendliness of smartphone apps is mainly measured based on single-source analysis (using one data-set). It is not a comprehensive measurement regarding the actual privacy risks of apps (Hatamian et al.2019). Another limitation is grounded in the complexity of these studies, that make it difficult to reach a high number of participants. While some of the weaknesses and limitations of Hatamian et al. (2018) were addressed in his future work by a multi-source analysis. The recent papers (Hatamian et al. (2019a), Hatamian et al. (2021)) also exhibit limitations on their own. The main focus in permission analysis are on dangerous permissions (see Table 8 for amdefinition). However, Hatamian et al. (2021) rightly points out that non-dangerous permissions also pose a risk to user's privacy. They can get exploited to profile users. He gives as an particular example, GET TASKS permissions can reveal sensitive information about which apps are being used by the user. Another main limitation is that all the chosen works

	Ex-ante transparency		Ex-post transparency	
	Permission Manifest Analysis	Privacy Policy Analysis	Permission Usage Analysis / App Behavior Analysis	User Reviews Analysis
Hatamian et al. 2018	☐	☐	☑	☐
Hatamian et al. 2019	☑	☑	☑	☑
Hatamian et al. 2021	☑	☑	☑	☐
Bock et al. 2021	☐	☐	☑	☐
Enck et al. 2014	☐	☐	☑	☐
Bal et al. 2014	☐	☐	☑	☐
Amini et al. 2013, 2014	☐	☐	☑	☐

Table 7: Used analysis methods

are limited to the Android platform only. Regardless of the choice of the research area, currently the A3, as well as the other tools cannot be completely applied to other smartphone platforms (e.g. iOS). This is due to the different technical conditions of the operating systems. While Hatamian et al. (2021) believes that the results from a privacy policy analysis can be transferred to IOS platforms, because the published privacy policies under investigation are generally identical for all apps, independent of the platform, this is not the case for all analysis types. To run a resource access pattern analysis on run-time the IOS device must get jailbreaked for example. This would again lead to limitations in regards to the completeness of the analyzed data set, since not all apps are available for iOS. It also has to get point out that the presented results can not be granted reproducibility. Since apps and also the platform itself are always evolving. Regular updated on the app or the privacy policies make static data snapshots outdated. To reproduce one must retrieve the older versions of apps, privacy policies, and various forks of the Android operating system to create an equal test bed. Additionally it can be expected that the documented app behavior deviates from the actual privacy-friendliness-level of apps in a real-life device usage scenario according to Hatamian et al. (2021). This is due to the following challenges in the preparation of a test environment: (a) apps can be subject to certain geographic installation only, (b) not ever app is compatible to run on rooted test devices, (c) conclusion of older test devises because of higher requirements (d) when apps demand registration with citizen data to run and (e) apps' documentation can be incomplete or in different languages. These challenges have to take into consideration when planning to conduct future research.

normal	dangerous	signature	signatureOrSystem
allow access to resources that are considered low-risk, and they are granted during installation of any package requesting them e.g. Internet, Bluetooth, vibrate, modify_audio_settings (Hatamian et al. 2021, Fig.2)	are required to access resources that are considered to be high-risk. In this case, the user must grant permission e.g. call logs, contact lists, location tracks, (Hatamian et al., 2021, Fig. 2)	grant access only to packages with the same author e.g.GET_Tasks, System_Alert-Window, request_install_packages (Hatamian et al. 2021, Fig. 2)	grant both packages with the same author and packages installed in the system receive permission to access specific resources

Table 8: 4 types of Permissions based on Hatamian et al. (2021)

4 Conclusion

To recognize and prevent the privacy threats from mobile applications, we looked into several tools

for identifying contradictions in privacy policies and supporting smartphone users to make privacy decisions by informing app permissions. Privacy policies are difficult to read and comprehend. However, they might violate the privacy of the mobile application users. It is crucial that the user understand the privacy policies and increase the awareness of mobile data collection. In this paper, We explained the process of dealing with extracting privacy policies and compared the different tools. We found out that PurPliance has the best performance on detecting the contradictions and inconsistencies in purposes between privacy policies and app's data transfer. Since there is a discrepancy between privacy policies and real application behavior, we pay attention to the tools that can identify data collections through app permissions. We discovered that app permissions such as location, camera and contacts these important personal information will be provided by the user when many of them are unaware of it. Permissions and policy assessments are required to be visualized to raise user's awareness.One hypothetical solution is to use an A3 tool to analyze android app behavior. Unfortunately, to best of our knowledge, there are no standardized tools available or ready to be used in the mobile ecosystem. Existing research have used either custom or have modified the general privacy risk assessment method of mobiles. Another limitation of these tools are that they can only be used on android applications, therefore, more attention can be paid to data collection from iOS.

References

Shahrihar Amini, Jialiu Lin, Jason I Hong, Janne Lindqvist, and Joy Zhang. 2013. Mobile application evaluation using automation and crowdsourcing.

Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. 2019. {PolicyLint}: Investigating internal privacy policy contradictions on google play. In *28th USENIX security symposium (USENIX security 19)*, pages 585–602.

Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. 2020. Actions speak louder than words:{Entity-Sensitive} privacy policy and data flow analysis with {PoliCheck}. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 985–1002.

Kfir Bar, Vered Zilberstein, Ido Ziv, Heli Baram, Nachum Dershowitz, Samuel Itzikowitz, and Eiran Vadim Harel. 2019. Semantic characteristics of schizophrenic speech. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 84–93, Minneapolis, Minnesota. Association for Computational Linguistics.

Sven Bock, Ashraf Ferdouse Chowdhury, and Nurul Momen. 2021. Partial consent: A study on user preference for informed consent. In *International Conference on Human-Computer Interaction*, pages 198–216. Springer.

Duc Bui, Yuan Yao, Kang G Shin, Jong-Min Choi, and Junbum Shin. 2021. Consistency analysis of data-usage purposes in mobile apps. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2824–2843.

Kirsten Crager and Anindya Maiti. 2017. Information leakage through mobile motion sensors: User awareness and concerns. In *Proceedings of the European Workshop on Usable Security (EuroUSEC)*.

Lorrie Faith Cranor, Pedro Giovanni Leon, and Blase Ur. 2016. A large-scale evaluation of us financial institutions' standardized privacy notices. *ACM Transactions on the Web (TWEB)*, 10(3):1–33.

William Enck, Peter Gilbert, Seungyeop Han, Vasant Tendulkar, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N Sheth. 2014. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)*, 32(2):1–29.

Alessandra Gorla, Ilaria Tavecchia, Florian Gross, and Andreas Zeller. 2014. Checking app behavior against app descriptions. In *Proceedings of the 36th international conference on software engineering*, pages 1025–1035.

Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. 2018. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 531–548.

Majid Hatamian. 2020. *A Multi-perspective Transparency and Privacy Enhancing Framework for Smartphone Apps*. Ph.D. thesis, Johann Wolfgang Goethe-Universität in Frankfurt am Main.

Majid Hatamian, Agnieszka Kitkowska, Jana Korunovska, and Sabrina Kirrane. 2018. "it's shocking!": Analysing the impact and reactions to the a3: Android apps behaviour analyser. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 198–215. Springer.

Majid Hatamian, Nurul Momen, Lothar Fritsch, and Kai Rannenberg. 2019a. A multilateral privacy impact analysis method for android apps. In *Annual Privacy Forum*, pages 87–106. Springer.

Majid Hatamian, Jetzabel Serna, and Kai Rannenberg. 2019b. Revealing the unrevealed: Mining smartphone users privacy perception on app markets. *Computers & Security*, 83:332–353.

Majid Hatamian, Jetzabel Serna, Kai Rannenberg, and Bodo Igler. 2017. Fair: Fuzzy alarming index rule for privacy analysis in smartphone apps. In *International Conference on Trust and Privacy in Digital Business*, pages 3–18. Springer.

Majid Hatamian, Samuel Wairimu, Nurul Momen, and Lothar Fritsch. 2021. A privacy and security analysis of early-deployed covid-19 contact tracing android apps. *Empirical software engineering*, 26(3):1–51.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Rahul Pandita, Xusheng Xiao, Wei Yang, William Enck, and Tao Xie. 2013. {WHYPER}: Towards automating risk assessment of mobile applications. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 527–542.

Zhengyang Qu, Vaibhav Rastogi, Xinyi Zhang, Yan Chen, Tiantian Zhu, and Zhong Chen. 2014. Autocog: Measuring the description-to-permission fidelity in android applications. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1354–1365.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. 2015. Android permissions remystified: A field study on contextual integrity. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 499–514.

Le Yu, Xiapu Luo, Jiachi Chen, Hao Zhou, Tao Zhang, Henry Chang, and Hareton KN Leung. 2021. Ppchecker: Towards accessing the trustworthiness of android apps' privacy policies. *IEEE Transactions on Software Engineering*, 47(02):221–242.

Le Yu, Xiapu Luo, Xule Liu, and Tao Zhang. 2016. Can we trust the privacy policies of android apps? In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 538–549. IEEE.

Sebastian Zimmeck and Steven M Bellovin. 2014. Privee: An architecture for automatically analyzing web privacy policies. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 1–16.

Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven Bellovin, and Joel Reidenberg. 2016. Automated analysis of privacy requirements for mobile apps. In *2016 AAAI Fall Symposium Series*.