Mobile Application Privacy Risk Assessments from User-authored Scenarios

Tianjian Huang, Vaishnavi Kaulagi, Mitra Bokaei Hosseini[†], Travis Breaux
School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States
[†]*Department of Computer Science, University of Texas at San Antonio, San Antonio, Texas, United States*
{*thuang2,vkaulagi,breaux*}*@cmu.edu, mitra.bokaeihosseini@utsa.edu*

***Abstract*—Mobile applications (apps) provide users valuable benefits at the risk of exposing users to privacy harms. Improving privacy in mobile apps faces several challenges, in particular, that many apps are developed by low resourced software development teams, such as end-user programmers or in startups. In addition, privacy risks are primarily known to users, which can make it difficult for developers to prioritize privacy for sensitive data. In this paper, we introduce a novel, lightweight method that allows app developers to elicit scenarios and privacy risk scores from users directly using only an app screenshot. The technique relies on named entity recognition (NER) to identify information types in user-authored scenarios, which are then fed in real-time to a privacy risk survey that users complete. The best-performing NER model predicts information types with a weighted average precision of 0.70 and recall of 0.72, after post-processing to remove false positives. The model was trained on a labeled 300-scenario corpus, and evaluated in an end-to-end evaluation using an additional 203 scenarios yielding 2,338 user-provided privacy risk scores. Finally, we discuss how developers can use the risk scores to prioritize, select and apply privacy design strategies in the context of four user-authored scenarios.**

***Index Terms*—requirements, scenarios, entity extraction, privacy, risk**

I. Introduction

Mobile applications (apps) have transformed how people use software by providing popular sources of on-demand entertainment, shopping, travel and business services, among others. Over 80% of US and EU adults use mobile apps [23], [55], on average for over four hours a day [19]. While mobile apps provide many benefits, they also introduce privacy risks due to the sensitivity of information collected, including information about social relationships, real-time location, finances and health. Privacy-by-design is legally required by laws such as the California Consumer Protection Act (CCPA) and General Data Protection Regulation (GDPR), however, these laws are limited when companies have fewer than 250 employees, or who earn less than $25 million in annual gross revenue. As a result, startup and small companies may lack the robust practices required to implement privacy programs.

Mobile app developing companies are often composed of a small number of developers with limited resources. Coupled with demanding development timelines, this makes early design and risk analysis challenging or impossible [43], [50]. Shilton & Greene found that app store approval processes and platform permission systems, which are frequently limited to a small number of information types (e.g., contacts, photos, location, etc.), motivate app developers to consider privacy-by-design [63]. Studies reveal that most developers need more formal knowledge of privacy and security practices [9] and that they try to integrate privacy features into software design without much understanding [62]. Oetzel et al. argue that developers require significant effort to estimate privacy risk from the end-user perspective [54]. Bhatia & Breaux show that users can estimate their privacy risk, despite lacking design knowledge and despite the privacy paradox, wherein users accept privacy risk in exchange for benefits provided by app [13]. Without privacy-by-design tools, developers may need to rework apps after apps they are in use by users.

In this paper, we describe a lightweight, semi-automated method that developers can use to identify sensitive information types, including types that are not controlled by mobile app permission systems, to be used as inputs to privacy-by-design strategies after the initial mobile app has been developed. To illustrate in Figure 1, a developer using the method identifies screenshots from their app's user interfaces and then seeks feedback from users in the form of user-authored scenarios describing how those users interact with the selected screens. The feedback is collected using fully automated online tools and then processed using named entity recognition (NER) to identify information types described by those users. Next, the information types are presented automatically to the same users to elicit their perceived privacy risk score for each information type. The developer can then use the privacy risk scores to review the overall app design and select from eight privacy design strategies defined by Hoepman [32].

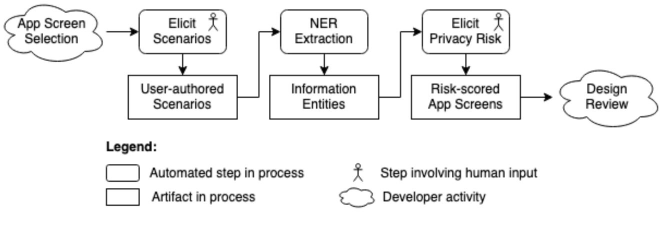


Fig. 1. Scenario-driven Privacy Risk Assessment

The contributions of this paper are: (1) the design of the surveys to collect scenarios and privacy risk scores; (2) a 300-scenario corpus with labeled information types describing 218 unique mobile apps; (3) an empirical evaluation of two

NER models (BERT and CRF); (4) an end-to-end evaluation performed using an additional 203 scenarios yielding 2,338 user-provided privacy risk scores; and (5) a discussion of how a developer can use the method results to select privacy design strategies based on a user's privacy priorities.

The remainder of this paper is organized as follows: in Section II we review background and related work; in Section III we introduce our research method to discover and simulate the technique with real users; in Section IV we report our results; in Section V we discuss results; in Section VI we review threats to validity; and we conclude with future work in Section VII.

II. Background and Related Work

We now review related work on scenarios, privacy attitudes and measuring privacy risk, before reviewing work on named entity recognition (NER) applied to extracting requirements-related information from software artifacts.

A. Scenarios and User Stories

Scenarios describe a concrete invocation of a system through a sequence of steps, often from a user's perspective [72], [73], and are widely used in software engineering [82], as well as human-computer interaction [48], and organizational process design [7]. Scenarios have been used to elicit, analyze, and validate requirements [2], [7], [33], including quality requirements for security [31], resiliency [26] and safety [4]. Scenarios can surface requirements unforeseen by business analysts and improve requirements alignment with users [57], and can be used to create more robust domain models [74]. Scenarios describe a system at different levels of abstraction [52], can be used in tracing requirements to software architecture [80] and code [42], [44], [52], [56], and in software testing [22]. When considering different stakeholder perspectives, scenarios can illustrate areas where value-conflicts arise [76]. Scenarios can be combined with personas and goal modeling to identify conflicting requirements [8]. Techniques exist to validate the scenario syntax and grammar using templates and rule-based verification [78], to identify missing steps [40], and to compute scenario similarity [5]. Scenarios can be used to validate formal models by challenging model assumptions [35], [72], to semi-automatically derive use-cases [4], and to identify functional requirements from scenario steps [36].

Whereas scenarios describe multiple steps, user stories are more concise and expressed using various templates [81], among which the Connextra format is most popular [45], i.e., as a <role>, I want <action>, so that <benefit>. User stories can be mapped to scenarios by elaborating on the *action* from the user's perspective. The general level of detail, which hides the underlying interactions with software, has made user stories popular in agile software development whereby the story summarizes one or more units of work in a iteration [39], [46]. When eliciting requirements from stakeholders, however, there are multiple what, how and why questions to investigate [57], which is why we chose to use scenarios instead of user stories in our method.

B. Privacy Attitudes and Risk

Privacy researchers have sought to understand why individuals share sensitive data with organizations that might misuse that data. Alan Westin introduced the Privacy Segmentation Index [83] through a series of surveys to segment individuals based on their relationship to privacy: *fundamentalists* are generally distrustful of organizations, *pragmatists* weigh the costs and benefits of trust, and the *unconcerned* are generally trustful of organizations. Acquisti and Grossklags studied the privacy paradox, in which user behaviors indicate a low value placed on privacy despite what they self-report [1]. Dupree et al. clustered users into five privacy behavior categories: fundamentalists, lazy experts, technicians, amateurs, and the marginally concerned [21]. These categories may explain why some users are more or less concerned about their privacy. Kang et al. found that Amazon Mechanical Turk (AMT) workers value anonymity and hiding information and had more privacy concerns than the general U.S. public [37].

Risk has been studied in marketing, psychology, and economics [69], with popular definitions focusing on a function of the likelihood and magnitude of an adverse event [70]. Marketing risk is a choice among multiple options based on the likelihood and desirability of the choice's consequences [11], whereas pyschological risk is an individual's willingness to participate in an activity [25], [69]. Kaplan and Garrick define economic risk as a function of probability and consequence, where the consequence is the measure of damage or harm [38]. While Cronk adapts economic risk to privacy [17], Bhatia and Breaux adapt psychological risk to an individual's willingness to share personal data [13], which they have studied in the context of vague and ambiguous data practices [14]. While users are known to rarely read privacy policies describing data practices [71], evidence shows users can estimate their privacy risk using data-specific prompts [13].

C. Natural Language Processing

In requirements engineering, natural language processing techniques have been used to extract important entites from text-based artifacts. Pudlitz et al. apply LSTMs and convolutional neural networks (CNNs) to extract system state descriptions from requirements specifications [58], and Siahaan et al. used named-entity recognition (NER) to extract hard- and soft-goals from online news sources [64]. In privacy and security, NER and transformer-based deep neural networks have been used to generate access control policies from user stories [29], and extract data-flow diagram elements from user stories [30], frequently focusing on the data subject, data type and data action. Casillo et al. apply CNNs to classify words in user stories as disclosure-related (e.g., access, share) [15]. Others have used part-of-speech-based rules to identify information types in privacy policies [12], and RNNs to extract penality clauses from regulations [6]. Sannier et al. have used regular expressions [60] and constituency and dependency parsing [68] to extract legal primitives from laws, which can then be used to query a legal knowledge base [67].

Social media posts [53] and mobile app reviews [27], [34] that describe user in-app experiences have been proposed as a source of requirements, including user opinions. These approaches employ sentiment analysis and typed dependencies, for example. Hatamian et al. analyze 812,899 app reviews from the top 10 apps in each of 20 categories on Google Play (200 apps total), and found less than 2,500 reviews (or 0.31%) of all reviews raise privacy concerns [28]. Topics raised in privacy concerns include tracking and spyware, phishing, unintended disclosures, targeted ads, and spam. However, others note that these approaches can be noisy and hard to replicate, with measured recall as low as 0.34 and 0.44 for two popular app-review mining approaches [18]. Moreover, app reviews and social media rely on users reaching a level of undesirable frustration before they raise such concerns publicly, which developers should want to avoid. Thus, we propose a method that invites users to directly comment on specific app screens, which developers can deploy with little manual intervention.

III. Method and Approach

The privacy risk assessment method shown in Figure 1 consists of three method steps: step (M1) to elicit user-authored scenarios; step (M2) to apply NER to extract information types; and step (M3) to elicit privacy risk scores from users using the extracted types. In this section, we describe two research study designs, including a formative study to collect scenarios using step M1 for training and testing the NER models, and a summative study to evaluate the end-to-end risk assessment tool consisting of steps M1-M3 using the best performing NER model from the formative study.

The research is guided by the following research questions:

RQ1: How well can contemporary NLP models extract personal information type entities?

RQ2: To what extent can users differentiate the sensitivity of personal information described in their self-authored scenarios?

RQ3: Are there fundamental differences in how users express their risk perceptions?

The RQ1 focuses on the feasibility of tools to automatically extract personal information types from scenarios. The RQ2 and RQ3 focus on whether the method can help developers distinguish between more- or less-sensitive information types.

We now described the three method steps and how we propose to answer the above research questions.

A. Eliciting User-authored Scenarios

In method step M1, mobile app users write scenarios about their interactions with the app's user interface. We study this step using the following survey steps: step (S1) the user chooses an app by identifying the mobile app URL from the Google Play or Apple App store; step (S2) the user identifies three types of personal information that the app collects, uses, or shares and they rate the privacy risk of sharing that information; step (S3) using a mobile device, the user takes a screenshot of the chosen app and redacts any personal information from the screenshot before uploading the screenshot to a server; and step (S4) while viewing the chosen screenshot, the user responds to the following prompt: "Write a brief 150-word minimum description of how you use this screen, including: (1) describe the goals you want to achieve through the screen; (2) your interactions with this screen to achieve your goals; and (3) the information that is used by the app to support this screen." Users are asked to avoid choosing the app's profile page, settings page, homepage or login page.

In survey step S3, the user is presented with a QR code that links to a web page where the user can choose a screenshot from their phone and proceed to draw redaction boxes on the screenshot to block out personal information. When the user is ready, only the redacted image is uploaded to the server. Figure 2 presents an example screenshot from the Pinterest app with one redaction box, numbered one. After the screenshot has been uploaded, the user is asked to describe the redacted information in general terms. The one redaction (lower-right corner) was described as the user's profile picture.

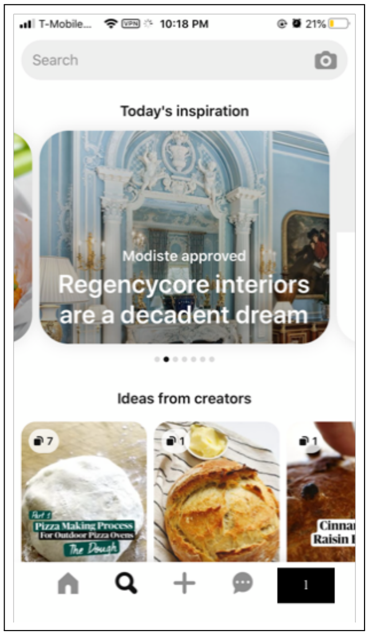


Fig. 2. Example of a User-Redacted Screenshot

In practice, a developer can select specific screens in their mobile app and use their user's contact information, such as e-mail addresses or in-app notifications, to recruit users to write scenarios. In that setting, the developer may only use step S4 in connection with the developer's selected screen. The developer can choose screens to increase privacy risk assessment coverage across multiple app screens. In our research, we bootstrap the developer process by allowing users to choose their own screen in step S1, while focusing their attention on privacy sensitivity in step S2, and we validated the authenticity of the user-app relationship using the screenshot in step S3.

Scenarios were collected using a survey[1] published on the

[1] https://github.com/cmu-relab/scenario_risk_scores

Amazon Mechanical Turk platform, wherein workers must have completed 5,000 Human Intelligence Tasks (HITs), have an approval rating greater than 97%, and be located in the United States. Workers were compensated $4.00 USD upon completion of the survey. As part of our protocol to protect human subjects, workers are required to provide informed consent before participating in the survey, and the study is monitored by our Institutional Review Board (IRB).

B. Training the NER models

In method step M2, a named-entity recognition (NER) model is used to label information types in user-authored scenarios. We use NER models because they support multi-word labeling tasks (i.e., information type phrases span multiple words), and because they do not impose limits on sentence length. When processing unseen texts, NER models generalize better than rule-based methods that use syntactic features provided by typed dependency and constituency parsers.

We evaluate two NER models: (1) a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model [20] that was pre-trained on the English version of the standard CoNLL-2003 Named Entity Recognition dataset [77]; and (2) a Conditional Random Field (CRF) model [49]. BERT-based models have shown promise in a variety of classical NLP tasks, including part-of-speech tagging, NER, semantic role labeling, and co-reference resolution [75]. A BERT variant, called RoBERTa [88], is used as the standard baseline in the Multilingual Complex Named-Entity Recognition (MultiCoNER) competition, which began in 2022 [47]. The classic BERT model is first pre-trained using unsupervised machine learning over a 3.3 billion word corpus [20]. Next, the model is fine-tuned for a specific task, such as NER, using a smaller corpus consisting of thousands of training instances. Conditional random fields are undirected graphical models that have efficient procedures for training and inference [49]. The CRF has been the standard NER model in the Stanford CoreNLP framework since 2005 [24].

In this work, the NER models are trained and evaluated using a 300 scenario corpus acquired in step M1. No additional steps were taken to pre-process the scenarios, e.g., by correcting spelling or grammar, removing special characters, or word stemming. To build the training dataset, the first and second authors coded sample scenarios in four rounds using coding theory [59]. In round one, both authors coded a random sample of 10 scenarios to identify information types, which resulted in Cohen's Kappa [16] of 0.297, which is fair agreement [41]. The authors next met to discuss disagreements and identify heuristics to clarify boundary and edge cases. In round two, the authors coded a new random sample of 10 new scenarios using the new heuristics, which resulted in a Kappa of 0.575. The authors reconvened, examined disagreements and developed the following coding frame, consisting of three sub-codes that predict the three author representations of information in their scenarios:

1) **Simple** (SIM): The information type appears as a noun phrase, including adjectives and excluding determiners, e.g., "purchase," "pinned playlists," "favorite subreddits"
2) **Complex** (COM): The information type appears as a simple type including a clause, e.g., a prepositional or verb phrase in "money in my account," "deal of the day," "movies that I want to purchase," and conjunctions "data about my health and activity"
3) **Question** (QUE): The information type appears as a question, including wh-clauses, e.g., "What I need and want to check for that day," and conditional clauses, e.g., "whether there is an accident"

In round three, the authors re-coded the second sample using the new coding frame to reach a Kappa of 0.694. In the fourth round, the authors coded a new random sample of 10 general scenarios using the new coding frame to yield a Kappa of 0.775, which is substantial agreement [41]. The remaining scenarios were then coded by the first author using the coding frame and accompanying heuristics. The final corpus consists of 4,163 information type phrases, including 3,524 SIM phrases, 302 COM phrases, and 337 QUE phrases. Overall, 1,881 phrases are lexically unique. This corpus was labeled using the Doccano[2] tool supporting overlapping annotations.

After labeling the corpus, the corpus was transformed into word tokens using SpaCy[3] v3.5 for word and sentence-level tokenization. Next, the annotations were converted to BIO-format, wherein a labeled word is the (B)eginning, (I)nside or (O)utside of the entity annotation, by aligning the character-level indices provided by Doccano with the token-level indices provided by SpaCy. Because the BIO-format does not support learning from overlapping annotations, we removed the shortest annotation in any two overlapping annotations (e.g., where a question annotation overlaps one or more simple annotation, the question annotation was preserved.) This strategy increased instances of the rarer COM and QUE annotations, which tend to have longer character spans than SIM annotations.

Table I presents an example sequence of labeled tokens, one per line, from a user-authored scenario sentence to illustrate the labels assigned by the coders (Expected) and the labels predicted by a NER model (Predicted). The SpaCy tokenizer standardizes the lexical representation of word punctuation, e.g., by separating contractions (see lines 2 and 16). In this example, the coder coded "recipe" with the SIM label on line 6, and the NER model predicted that the word sequence "pasta dinner recipe" should be labeled SIM. This prediction is consistent with noun phrases where preceding nouns and adjectives are included in a compound noun. The coder also identified lines 11-18 as a COM label, as well as lines 14-18 as a QUE label, however, the rules for converting overlapping annotations to non-overlapping annotations requires choosing the code with the longest span, which was the COM label. The NER model, on the other hand, optimizes by predicting the best sequence among possibilities: that "phrase" (line 11)

[2] https://github.com/doccano/doccano

[3] https://spacy.io/

could be a SIM label, or part of a COM label (see lines 12-13) and that "what I'm looking for" could be a QUE label.

Line	Token	Expected	Predicted
1	I	O	O
2	'll	O	O
3	type	O	O
4	pasta	O	B-SIM
5	dinner	O	I-SIM
6	recipe	B-SIM	I-SIM
7	,	O	O
8	then	O	O
9	select	O	O
10	the	O	O
11	phrase	B-COM	B-SIM
12	that	I-COM	B-COM
13	matches	I-COM	I-COM
14	what	I-COM	B-QUE
15	I	I-COM	I-QUE
16	'm	I-COM	I-QUE
17	looking	I-COM	I-QUE
18	for	I-COM	I-QUE
19	.	O	O

TABLE I
EXAMPLE OF EXPECTED AND PREDICTED WORD LABELS

We fine-tuned the BERT model using the Hugging Face[4] API v4.26.0 and an 80/10/10 dataset split for training, validation and testing, respectively. The features are the word tokens and BIO-tags. The hyper-parameters consist of 10 training epochs, a learning rate of 2e-5, and weighted decay of 0.1.

We trained the CRF model using the SciKit Learn CRF Suite[5] v0.3 and an 80/10/10 dataset split for training, validation and testing, respectively. We first performed a randomized grid search using the validation set to identify optimal c_1 and c_2 parameters. The CRF model was then configured using the optimal hyper-parameters: algorithm = "lbfgs", c1 = 0.33596, c2 = 0.05948, max-iterations = 100 and all-possible-transitions = false. Unlike BERT-based models, where the features are learned during pre-training, CRF-based models require features to be defined by the user. The following eight features were used for each word in the corpus: (a) the lowercase word, (b) the word stemmed by 3 characters, (c) the word stemmed by 2 characters, (d) if the word is all upper case, (e) if the word is a title word, (f) if the word is a number, (g) the full part-of-speech (POS) tag obtained by SpaCy, and (h) the first two characters of the POS tag.

The RQ1 is answered by evaluating the NER model's precision, recall and F-1 scores: for true positives (TP), false positives (FP) and false negatives (FN), the *precision* is equal to TP / (TP + FP); *recall* is equal to TP / (TP + FN); *F-1* is the harmonic mean and equal to 2TP / (2TP + FP + FN); and the *support* is the number of TP entities represented in the calculations for the table row. The unit of analysis in these calculations is phrase-level, wherein a true positive occurs when a predicted phrase boundary exactly matches an expected phrase boundary with the correct sub-code, i.e., the BIO codes and the sub-code for simple (SIM), complex (COM) or question (QUE) must match, exactly. An alternative unit of analysis is word-level, in which a true positive occurs when a predicted word label has the same BIO code and same sub-code as an expected word label. The *sklearn.metrics* framework supports word-level evaluations.

Word-level metrics report higher precision and recall rates, because the distribution of word-level BIO codes in NER tasks is skewed toward the outside "O" code. Phrase-level metrics do not count outside codes as true positives, and thus are more conservative. Thus, word-level metrics lead to over-confidence by under-estimating the quality of extracted entities (e.g., word-level metrics accept partial matches as true positives).

To calculate phrase-level metrics, we use the *seqeval.metrics* framework developed for the CoNLL-2000 shared task [51]. The BERT and CRF models were each trained using the same 10 randomly selected 80/10/10 splits for training, validation and testing, respectively. Because the BERT-based model uses a sentence-level encoder-decoder architecture, the splits were performed over scenarios and not sentences. The comparative evaluation is based on the weighted average of the weighted averages for precision, recall and F-1 scores. We report the model evaluation in Section IV-A.

During an error analysis of false positives after testing, a post-processing algorithm was developed and applied to both NER model results to remove false positives produced by the models. The algorithm analyzes the lexeme and POS tags obtained using the SpaCy standard POS-tag parser and the following rules:

- Remove phrases ending DT, CC or $PRP (e.g., the, a, and, or, or your)
- Remove phrases beginning with POS, 'of' or CC (e.g., my, of, and)
- Remove phrases of word-length one that have no NN nor end with VBG
- Remove phrases of word-length two that have no NN nor end with VBG

In some instances, the above post-processing removes incomplete, but correct labels. We chose to remove the labeled phrases from the output as opposed to correcting the incomplete labels, which is a topic for future work. The post-processing algorithm is applied to the best-performing model used to predict information types for method step M3, which we describe next.

C. Eliciting Privacy Risk Scores

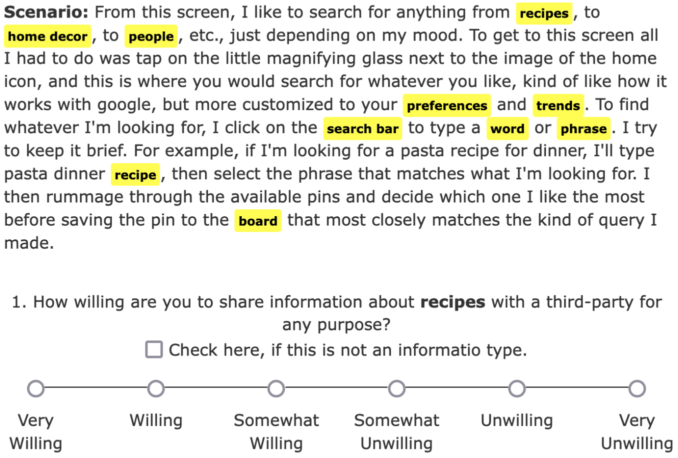
In method step M3, information type entities extracted from the user-authored scenarios by the best performing NER model in step M2 are immediately presented to the scenario authors in a second follow-on survey (see Figure 3). The authors see their original scenario with the predicted information types highlighted in yellow and the authors are asked for each type, "How willing are you to share [information type] with a third party for any purpose?" where the bracketed phrase is replaced by one of the highlighted types. If the bracketed phrase is not an information type (i.e., the NER produced a false positive), then the user can check a box indicating the error. Otherwise, the user responds using a six-point semantic scale adapted

[4] https://huggingface.co/dslim/bert-base-NER

[5] https://sklearn-crfsuite.readthedocs.io/en/latest/

from Bhatia et al. [13] with the scale anchors labeled Very Willing, Willing, Somewhat Willing, Somewhat Unwilling, Unwilling, and Very Unwilling.

Bhatia et al. defined privacy risk as the willingness to share one's personal information with a third party. As the person's perceived risk increases, the person is less willing to share their data. They note that willingness to share can be affected by multiple factors, including a person's perceived benefits of sharing, the type of information being sharing, with whom the information is shared and for what purpose. Paul Slovic in economics discovered that as perceived benefits increase, the perceived risk decreases across multiple activities [69]. In this survey, we assume that users are aware of their app's benefits, and we set the risk to *sharing the extracted information type with a nameless third-party for any purpose*, which Bhatia et al. describe as the highest level [13].



Scenario: From this screen, I like to search for anything from **recipes**, to **home decor**, to **people**, etc., just depending on my mood. To get to this screen all I had to do was tap on the little magnifying glass next to the image of the home icon, and this is where you would search for whatever you like, kind of like how it works with google, but more customized to your **preferences** and **trends**. To find whatever I'm looking for, I click on the **search bar** to type a **word** or **phrase**. I try to keep it brief. For example, if I'm looking for a pasta recipe for dinner, I'll type pasta dinner **recipe**, then select the phrase that matches what I'm looking for. I then rummage through the available pins and decide which one I like the most before saving the pin to the **board** that most closely matches the kind of query I made.

1. How willing are you to share information about **recipes** with a third-party for any purpose?

□ Check here, if this is not an informatio type.

○ Very Willing ○ Willing ○ Somewhat Willing ○ Somewhat Unwilling ○ Unwilling ○ Very Unwilling

Fig. 3. Example of a Privacy Risk Survey

Based on prior work to study privacy personas [21] and user segmentation based on privacy attitudes [37] (see Section II), we investigate RQ2 and RQ3 by analyzing scale usage by users in their privacy risk assessments. Kahneman and Tversky identified fundamental biases exhibited by people when they make estimates, such as when they estimate their privacy risk [79], including: *availability*, which occurs when a recent event biases an estimate (e.g., a significant data breach reported in the news, or a recent privacy harm experienced by the estimator), and *adjustment* and *anchoring*, which occurs when subsequent estimates are made in relation to a prior estimate. For example, in the response scale shown in Figure 3, respondents may skew their responses to the left or right based on recent events, or they may choose an overall attitude toward privacy (e.g., align with privacy fundamentalists or privacy unconcerned) and then choose subsequent ratings to fulfill that alignment. We investigate scale usage by examining the nominal ratings distribution by a single user and across users to identify risk perception patterns that can inform how to interpret responses. We do not study biases, such as availability, nor do we test whether scale usage aligns with a user's privacy behavior (e.g., the privacy paradox [1]).

This survey is evaluated in an end-to-end integration of the user-authored scenario survey from step M1, the best performing NER model from step M2, and the privacy risk survey from step M3. The integrated method is published on the AMT platform, wherein workers must have completed 5,000 Human Intelligence Tasks (HITs), have an approval rating greater than 97%, and be located in the United States. Workers were compensated $4.00 USD upon completion of the survey. As part of our protocol to protect human subjects, workers are required to provide informed consent before participating in the survey, and this study is monitored by our Institutional Review Board (IRB).

IV. Results

We now review the named entity recognition (NER) model and the end-to-end risk elicitation method results.

A. Formative Study Results

The BERT- and CRF-based models were trained and evaluated using the 300-scenario corpus collected using the survey method described in Section III-A. Among the 300 user-authored scenarios, 106 scenarios cover 21 App Store categories, and 194 scenarios cover 24 Play Store categories, representing a broad and diverse category set, including Books & Reference, Education, Finance, Health & Fitness, Lifestyle, Navigation, News, Photo & Video, Productivity, Shopping, Social Networking, Sports, Travel and Weather, among others. The corpus is described in Table II, which presents the number of Apple App store apps, Google Play store apps and overall unique apps, as well as, the total number of scenarios, unique authors, sentences and information types coded as simple (SIM), complex (COM) and question (QUE) entities according to the definitions in Section III-B.

Attribute	**General**	**Privacy**	**Total**
Apple App	38	68	106
Google Play	62	132	194
Unique Apps	84	134	218
Scenarios	100	200	300
Authors	88	144	232
Sentences	810	1,616	2,426
Entity SIM	1,115	2,409	3,524
Entity COM	94	243	337
Entity QUE	123	179	302
Total Entities	**1,332**	**2,831**	**4,163**

TABLE II
300-Scenario Corpus Attributes and Frequencies

Table III presents the BERT-based and the CRF-based NER evaluation results using the *seqeval.metrics* for the weighted average precision, recall and F-1 defined in Section III-B, weighted over 10 randomly sampled subsets of training, validation and testing data. In Table III, the bold values represent the best average measurements. To answer RQ1 about the quality of NER model performance, we observe that the BERT-based model performed better on recall for QUE and SIM, and marginally better on recall for COM. The CRF-based model performs better on precision for all types (COM, QUE, SIM). With the higher performance on recall, we selected

	BERT-based NER Model				CRF-based NER Model			
Attribute	**Precision**	**Recall**	**F-1**	**Support**	**Precision**	**Recall**	**F-1**	**Support**
Entity COM	0.04	**0.07**	0.05	31	**0.11**	0.05	**0.06**	32
Entity QUE	0.37	**0.62**	0.46	32	**0.62**	0.41	**0.49**	34
Entity SIM	0.68	**0.79**	0.**73**	346	**0.71**	0.67	0.69	356
Weighted Avg	0.61	**0.72**	**0.66**	410	**0.66**	0.61	0.63	422

TABLE III
EVALUATION OF BERT-BASED AND CRF-BASED NER MODELS

the mid-performing or average model in the best-performing model class (BERT), which has a weighted precision = 0.62, recall = 0.73, and F-1 = 0.66.

In Section III-B, we describe post-processing rules that are applied to reduce false positives in the model prediction. When this technique is applied to the selected BERT-based model after testing, it yields a final weighted average precision of 0.70, which is an increase from 0.62. While high precision reduces the false positives encountered by scenario authors when asked to score privacy risk in step M3, we also value high recall for better generalizability.

B. *Summative Study Results*

In the end-to-end, summative study, we collected an additional 203 user-authored scenarios while employing the average BERT-based NER model to identify information type entities and asked users to rate the privacy risk of sharing each entity (see Section III-C). Similar to the corpus described in Table II, the attributes of the 203-scenario corpus are: 71 Apple App, 132 Google Play, 148 Unique Apps, 121 Unique Authors, and 1,592 Sentences. In addition, the BERT-based NER model identified 2,605 information type entities, among which users reported a total of 267 false positives to yield a user-perceived precision of 0.89. Each scenario yielded an average of 12.8 entities per scenario. Among the scenarios described, 53% of users report using the app daily, and 38% weekly.

The ratings distribution for all 2,338 user-reported risk ratings appears evenly distributed across the scale. The following frequencies of user ratings correspond to the labeled scale anchors with the scale score in parentheses: 473 ratings for Very Willing (score 0.0), 436 ratings for Willing (score 1), 478 ratings for Somewhat Willing (score 2), 353 ratings for Somewhat Unwilling (score 3), 295 ratings for Unwilling (score 4), and 303 ratings for Very Unwilling (score 5). Higher scores (3 and above) correspond to greater privacy risk perception, and lower scores (2 and below) correspond to lower privacy risk perception [13]. In response to RQ2, we observe that users do discriminate privacy risk between different information types by making full use of the scale.

For the Apple app categories with more than ten apps represented, the average privacy risk score for Finance (12 apps) was 3.13, which is higher risk than the average for Health & Fitness (29 apps) at 1.21. The Google app categories with ten or more apps were scored on average as follows: Communications (10 apps) at 3.24, Food & Drink (16 apps) 2.80, Health & Fitness (17 apps) 1.55, and Social (19 apps) at 2.72. The overall average Apple app privacy risk score was 1.92, and the overall average Google Play app privacy risk score was 2.35.

Users rated the following information type entities as Very Willing and Willing to share: age, gender, height, weight, usage data, streaming and download quality, fitness and diet goals, food calories, subscriptions, orders, charges, and purchases. Among entities rated as Unwilling and Very Unwilling to share, users rated: payment information, including checking account number, location, email address, phone number, username, contact lists, and conversations.

Users perceive privacy risk differently for the same information type across apps, which partially answers RQ3. In Table IV, we present sentences from different scenarios that contain the phrase "phone number," in addition to the privacy risk score distribution for all information type entities scored by the user for this app. The score that was chosen for phone number is presented in parentheses. The table illustrates how users may integrate app usage context into their reported scores, e.g., "I can also use this screen to add new friends by searching for their usernames or **phone number**," which this user reported being *willing* to share this phone number with third parties. Another user reports being *very unwilling* to share phone number with third parties: "The app uses my contacts' names, **phone numbers**, and profile pictures to support this screen." Unlike the second context, the first context illustrates a user who discloses a friend's phone number through a user directory to identify friends, which may associate the identifier with properties of information found in a public directory, thus lowering the perceived privacy risk.

Table V presents the scale utilization in three categories: users who only used ratings on the Very Willing to Somewhat Willing-side of the scale (VW to SW), users who only used ratings on the Somewhat Unwilling to Very Unwilling-side of the scale (SU to VU), and users who used a mixture of Willing and Unwilling responses (Mixed W/U) when rating information types found in their scenarios. The numbered columns represent the number of different response options chosen by the users, e.g., 33 users chose only one response option for *all the information types* that they rated, whereas 60 users chose among three response options. On average, each user rated 12.8 information types for privacy risk. The overall average privacy risk score was 2.19, which is slightly more than Somewhat Willing (score 2.0). Overall, the table indicates diversity of privacy perspectives, which further answers RQ3.

C. *Developer Use Cases*

In the privacy assessment workflow presented in Figure 1, the resulting risk-scored, user-authored scenarios are then presented to the developer. We envision the developer using this information in conjunction with privacy design strategies to make design decisions that improve privacy. Hoepman defines

Scenario ID	Scenario Text	VW	W	SW	SU	U	VU
MAS-R-4	I can also use this screen to add new friends by searching for their usernames or **phone number**.	0	0	0	0	(5)	3
MAS-R-5	If I ever get banned from the chat, my **phone number** will be permanently banned, and I'd have to create a new account in order to chat again.	5	0	(1)	1	7	0
MAS-R-34	There are many people who I will need to look up an address or **phone number** for and I can find that information here.	0	2	6	(3)	0	0
MAS-R-44	Important is things like medical instructions, reminders to pick up medications, peoples **phone numbers** or doctors appointments or just general things to remember that are of priority to me.	2	2	5	0	0	(5)
MAS-R-47	I don't think both a email and **phone number** are required to use Zelle but I assume that it can be helpful.	0	1	(9)	0	0	0
MAS-R-48	The information that is used to support this screen is usually access to your email address, **phone number** or authentication device program that is on your phone that confirms that you are who you say you are.	0	0	0	0	1	(7)
MAS-R-64	The app uses my contacts' names, **phone numbers**, and profile pictures to support this screen.	0	1	1	0	2	(4)

TABLE IV
USER PERCEIVED PRIVACY RISK RATINGS AND DISTRIBUTIONS FOR SHARING PHONE NUMBER

Scale Segmentation	1	2	3	4	5	6	Total
VW to SW	22	27	16	0	0	0	**65**
SU to VU	11	9	8	0	0	0	**28**
Mixed W/U	0	16	36	35	17	6	**110**
Total Scenarios	**33**	**52**	**60**	**35**	**17**	**6**	**203**

TABLE V
SCALE UTILIZATION BY DIVERSITY OF RESPONSE OPTIONS CHOSEN

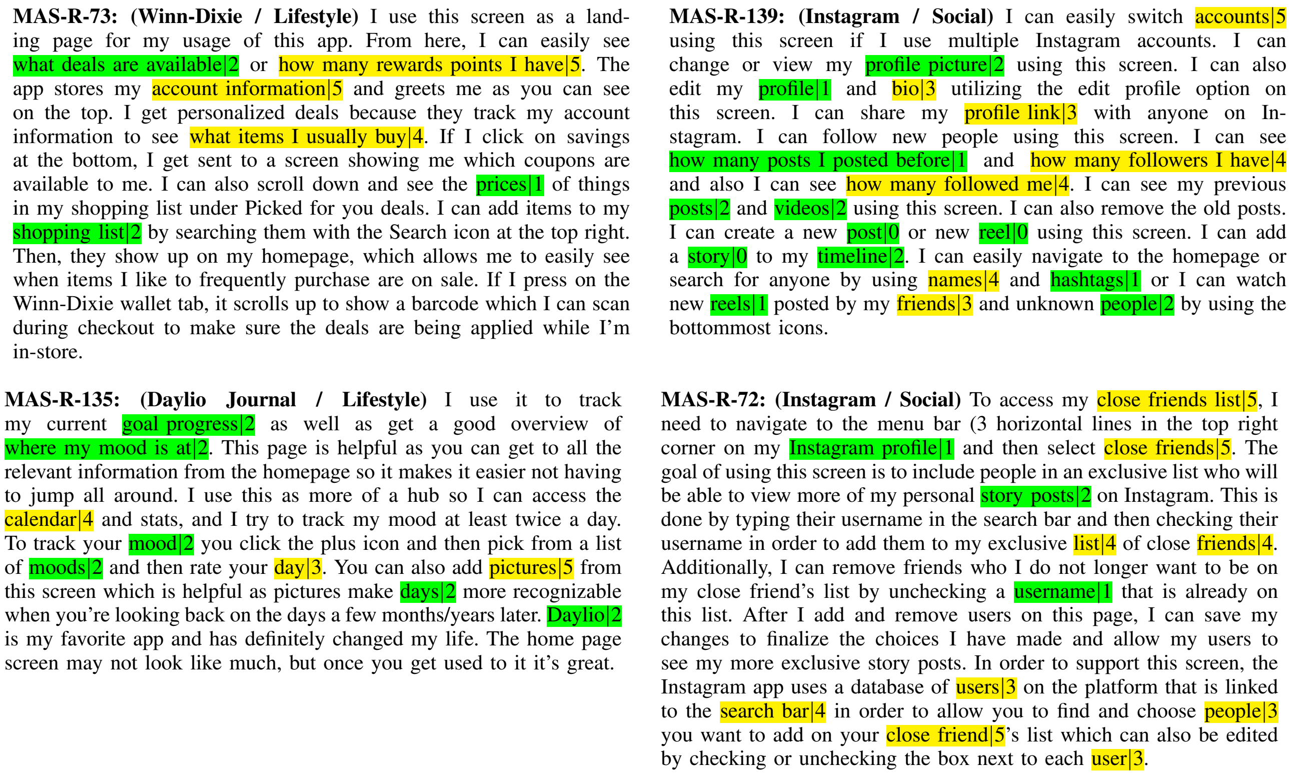
eight strategies (author's mnemonics in *italics*) [32]: *minimize* and *separate* the processing of personal data; *abstract* or remove detail from processable data; *hide* data by making it unobservable and unlinkable; *inform* data subjects about how their data is processed; provide data subjects with *control* over their data; and *enforce* commitments to process data in a privacy-sensitive manner.

We illustrate how a developer would use the risk-scored, user-authored scenarios by examining a few cases from the Lifestyle and Social app categories, which we chose because the risk scores in this category are diverse, and the category practices receive less regulatory guidance, unlike finance and healthcare. In the scenarios below, the predicted information types with risk scores of Somewhat Willing (score 2) or lower are highlighted in green, scores of Somewhat Unwilling (score 2) or higher are highlighted in yellow, and the risk score is presented to the right side of a vertical bar (e.g., email address|4). Throughout this section, we use the following risk labels: non-sensitive or low risk (scores 0-1), medium risk (scores 2-3), sensitive or high risk (scores 4-5).

We observed a few general guidelines for developers. First, if an information type has a risk score of Somewhat Unwilling (score 3) or higher, we recommend the developer *inform* users by disclosing this information type or its general category and the purpose for which it is processed in a privacy notice. If the type is especially sensitive and part of a broad category, the developer can disclose the type as an example of the category, e.g., "We share your account information, including your e-mail address, with third-parties for marketing purposes." If there is an action that the user takes within the app that can lead to disclosing a sensitive type, then we recommend using a just-in-time notice that appears just prior to the user taking this action [61]. Second, the developer should consider providing users *control* over sensitive types, which may include privacy settings to *hide* sensitive information from disclosure, or substitute less sensitive data for sensitive data (e.g., substitute an app-specific, weak identifier for a more universal, strong identifier, such as phone number), and to allow users to regenerate a new weak identifier, as needed to unlink their historical data from any third-party profiles constructed using this identifier. We now review specific sensitivities reported by users and discuss how the developer could apply the above strategies.

Figure 4 presents a Lifestyle app scenario MAS-R-73 about the Winn-Dixie app, which allows users to create grocery lists and accumulate shopping rewards for Winn-Dixie stores. To improve privacy, the developer should inform users through their privacy notices about how they process account information, reward points and items bought, because these data types are highest risk. The developer may choose to hide account information from any third-party sharing, because this is highest risk. If they want to link shopping list or items bought to third-party coupons, they may give users control via opt-in mechanisms for those features, because these are medium to lower risk. Finally, information about deals or prices in the user's area are perceived as low risk, in which case the developer may share this information, e.g., by allowing competitors to offer better coupons.

In Figure 4, a user describes the Daylio Journal app, a Lifestyle app in scenario MAS-R-135. In this app, a developer may emphasize calendar, date or day information and picture data in privacy notices, while also prohibiting the sharing of this data with others, because it is high risk. The user refers to mood in two risk-score contexts: where their mood is currently at (score 2) and a list of mood options (score 2) from which to label one's mood. Given mood is a medium risk, the developer may consider offering users just-in-time notices for when a user's mood changes or after a period of time to check whether the user has changed their sharing preferences [61]. As this scenario illustrates, how users write about and score their personal data can be subject to vagueness. The first "mood" refers to the user's mood, whereas the second "mood options" may only refer to the list, or it may also refer to the user's choice of their mood in the list. The developer should



MAS-R-73: (Winn-Dixie / Lifestyle) I use this screen as a landing page for my usage of this app. From here, I can easily see what deals are available|2 or how many rewards points I have|5. The app stores my account information|5 and greets me as you can see on the top. I get personalized deals because they track my account information to see what items I usually buy|4. If I click on savings at the bottom, I get sent to a screen showing me which coupons are available to me. I can also scroll down and see the prices|1 of things in my shopping list under Picked for you deals. I can add items to my shopping list|2 by searching them with the Search icon at the top right. Then, they show up on my homepage, which allows me to easily see when items I like to frequently purchase are on sale. If I press on the Winn-Dixie wallet tab, it scrolls up to show a barcode which I can scan during checkout to make sure the deals are being applied while I'm in-store.

MAS-R-139: (Instagram / Social) I can easily switch accounts|5 using this screen if I use multiple Instagram accounts. I can change or view my profile picture|2 using this screen. I can also edit my profile|1 and bio|3 utilizing the edit profile option on this screen. I can share my profile link|3 with anyone on Instagram. I can follow new people using this screen. I can see how many posts I posted before|1 and how many followers I have|4 and also I can see how many followed me|4. I can see my previous posts|2 and videos|2 using this screen. I can also remove the old posts. I can create a new post|0 or new reel|0 using this screen. I can add a story|0 to my timeline|2. I can easily navigate to the homepage or search for anyone by using names|4 and hashtags|1 or I can watch new reels|1 posted by my friends|3 and unknown people|2 by using the bottommost icons.

MAS-R-135: (Daylio Journal / Lifestyle) I use it to track my current goal progress|2 as well as get a good overview of where my mood is at|2. This page is helpful as you can get to all the relevant information from the homepage so it makes it easier not having to jump all around. I use this as more of a hub so I can access the calendar|4 and stats, and I try to track my mood at least twice a day. To track your mood|2 you click the plus icon and then pick from a list of moods|2 and then rate your day|3. You can also add pictures|5 from this screen which is helpful as pictures make days|2 more recognizable when you're looking back on the days a few months/years later. Daylio|2 is my favorite app and has definitely changed my life. The home page screen may not look like much, but once you get used to it it's great.

MAS-R-72: (Instagram / Social) To access my close friends list|5, I need to navigate to the menu bar (3 horizontal lines in the top right corner on my Instagram profile|1 and then select close friends|5. The goal of using this screen is to include people in an exclusive list who will be able to view more of my personal story posts|2 on Instagram. This is done by typing their username in the search bar and then checking their username in order to add them to my exclusive list|4 of close friends|4. Additionally, I can remove friends who I do not longer want to be on my close friend's list by unchecking a username|1 that is already on this list. After I add and remove users on this page, I can save my changes to finalize the choices I have made and allow my users to see my more exclusive story posts. In order to support this screen, the Instagram app uses a database of users|3 on the platform that is linked to the search bar|4 in order to allow you to find and choose people|3 you want to add on your close friend|5's list which can also be edited by checking or unchecking the box next to each user|3.

Fig. 4. Example 1 Risk-Labeled Scenario for Developer

be especially sensitive to these distinctions by considering contrasting examples when interpreting what users are scoring in their ratings.

In Figure 4, we present two scenarios authored by different users about the same app, Instagram. Developers benefit by reviewing scenarios from different users to infer categories of sensitive data. In both scenarios, for example, users rate their profiles as low risk, and their content as medium risk, including their profile picture, bio, story posts, and videos. Notably, scenario MAS-R-72 distinguishes close friends as high risk, whereas scenario MAS-R-139 scores their friends as medium risk, and the names of other users as high risk. This may indicate to the developer that, as the social proximity of friends to a user decreases, users perceive the privacy risk of that information as increasing, and that increasing specificity of information about friends increases privacy risk. In addition, the author of scenario MAS-R-139 views statistics about their posts as low risk, but statistics about who they follow and who follows them as high risk. This distinction challenges the conception that statistical or frequency data is lower risk. To address these nuances, the developer may rely more on *control* strategies to allow users to tailor their privacy.

V. Discussion

We now review the research questions in the context of the results. RQ1 asks whether contemporary NLP can be used to extract quality information type entities. The quality of extracted entities depends on the reliability of the coding frame. The first and second authors developed the frame over four coding rounds, in which the number of distinct entity labels was refined from ten to four and finally three categories, yielding an overall 0.775 Kappa score. When including post processing, the mid-performing, average BERT-based model has weighted precision 0.70 and recall 0.73, which is sufficiently high to integrate in an end-to-end assessment, wherein users reported a higher perceived precision of 0.89.

The ability to compare the BERT-based and CRF-based model performance required segmenting the training, testing and validation data on scenarios, because the BERT-based model can only be trained on sentence sequences, whereas the CRF-based model can be trained on word sequences. If we were to segment the corpus by sentences, which more evenly distributes label types across training, testing and validation data, then the CRF-based weighted metrics rise to 0.67 precision, 0.66 recall and 0.66 F-1, which is a 0.01, 0.06 and 0.03 increase over the scenario-trained model, respectively.

RQ2 asks to what extent users can differentiate the sensitivity of personal information in their scenarios. The scale utilization reported in Table V shows that users clearly distinguish between higher-risk information types, but also that some users may perceive risk according to broader conceptual frameworks described by privacy attitudes. For example, users who only use ratings from the Willing-side of the scale may be interpreted as privacy unconcerned, whereas users who only rated from the Unwilling-side of the scale may be interpreted as privacy fundamentalists. Mixed-use raters may be viewed as pragmatists. Prior research (e.g., contextual integrity [10],

[84], including refinements upon the privacy paradox [86]) has challenged the view that users are of only one type, however.

Finally, RQ3 asks whether users differ in their perceptions of privacy risk. Table IV compares the same information type across users and apps and demonstrates a difference in risk perception, in addition, scenarios written by different authors of the same app exhibit different risk scores for the same information type. We also observed that Apple device users rated information types on average 1.92 and Android device users rated information types on average 2.35, which is slightly below and above Somewhat Willing (score 2.0), respectively. Based on the Wilcoxon signed-rank test, this difference is statistically significant ($p = 0.000 < 0.05$, 95% CI, $N_1 = 808$, $N_2 = 808$). There are a few possible explanations: (1) Apple users selected apps less likely to expose privacy risks than Android users; (2) Apple users are generally more trusting of organizations (e.g., privacy unconcerned); and/or (3) Android users are more privacy aware, and thus reported more sensitive information types in their authored scenarios. Our study results do not indicate which explanation is likely, and thus this is a topic for future work.

VI. Threats to Validity

We now discuss threats to validity.

Construct validity is the correctness of operational measures used to collect data, build theory and report findings from the data [87], and the extent to which an observed measurement fits a theoretical construct [66]. To reduce this threat, the coders met in four rounds to identify discrepancies when labeling the dataset. The four rounds yielded an improved coding frame and heuristics as measured by Cohen's Kappa. However, the annotated types do not distinguish if the type is processed *within* or *outside* the mobile app system boundary. Thus, users may be asked to score risks on types that are not processed by the apps.

Internal validity is the extent to which measured variables cause observable effects in the data [87]. In study M2, users rate information types described in their scenarios using semantic scales. Rating scales are subject to cognitive biases, such as anchoring [79], in which subsequent responses are made relative to an initial response, called the anchor. In an extreme form of anchoring, respondents may choose one overall risk level and then apply that to each rated item. We analyzed the overall distribution of response levels (see Table V), and found that only 33/203 authors used only one level to score their privacy risk, and in fact more than half 110/203 rated a mixture of sensitive and non-sensitive types. We found that developers must consider the language context of the highlighted scenario phrase when interpreting risk scores, and they should consider scenarios from more than one author, before drawing conclusions from the scores.

External validity determines the scope of environmental phenomena or domain boundaries to which the theory and findings generalize [87]. The generalizability of the findings are limited to information types that users can observe. For example, a user's IMEA, which are internally used by apps as advertising identifiers, may not appear in scenarios, and thus would not be scored). In addition, if users are reluctant to share certain screens, then this research can miss sensitive information types. To mitigate this threat, we allow users to redact or mask areas of mobile app screens to hide sensitive information before submitting the screenshot. While we did observe sensitive screens, including screens with redacted gendered health data, financial account balances and precise routes used for exercise, we still agree that this limitation cannot be fully mitigated. That said, we believe the technique proposed in this paper is not limited by this limitation in the dataset. Finally, the model was trained on scenarios from 21 Apple and 24 Google Play store categories, representing a diverse dataset.

VII. Conclusion

In this paper, we describe an empirically validated method that we believe developers can use to elicit user-authored scenarios with user-reported privacy risk scores. The method is lightweight and requires minimal interaction from the developer. The developer need only select and share screenshots with users, and the technique reports the user's perceived risk scores for information types described in their scenarios. Thus, we believe the technique can be integrated into agile development processes that otherwise rely on limited documentation of requirements. Second, because privacy is about the data subject, the technique overcomes the limitations of using personas or other user representations to elicit requirements or approximate privacy risk. In contrast, the risk scores are provided directly by users in the context of how they use the app.

A disadvantage of the approach is that it is a post-production method that is deployable only after initial design and development work has been completed, and thus it cannot easily support privacy-by-design before an app is deployed. However, we believe that many mobile app developers postpone privacy considerations until after users have demonstrated an initial interest in using their apps. This is due in part to limited resources (e.g., too few developers, limited or no legal consultation) and competing priorities (e.g., producing a working prototype, pursuing early-stage startup investment). Thus, we see this technique as improving upon the current state of development until better frameworks and training become available.

In future work, we envision: (1) metrics to evaluate the quality of scenarios; (2) tools to guide scenario authors in writing higher quality scenarios; (3) classifying information types on whether they are inside/outside the app; (4) analyzing the scenario verbs to identify data actions that trace to software features; and (5) analyzing privacy policies to detect whether those high-risk types are mentioned.

Acknowledgment

This work was supported by NSF Award #2007298.

References

[1] A. Acquisti and J. Grossklags, “Privacy and rationality in individual decision making,” *IEEE Security & Privacy*, 3(1): 26-33, 2005.

[2] I.F. Alexander, N. Maiden, eds. *Scenarios, stories, use cases: through the systems development life-cycle.* John Wiley & Sons, 2005.

[3] G. Alexandron, M. Armoni, M. Gordon, D. Harel. “Scenario-based programming: Reducing the cognitive load, fostering abstract thinking.” *36th International Conference on Software Engineering*, pp. 311-320, 2014.

[4] K. Allenby, T. Kelly, “Deriving safety requirements using scenarios,” *Fifth IEEE International Symposium on Requirements Engineering*, pp. 228-235, 2001.

[5] T. A. Alspaugh, A. I. Anton, T. Barnes, B. W. Mott, “An integrated scenario management strategy,” *International Symposium on Requirements Engineering*, pp. 142-149, 1999.

[6] P. R. Anish, P. Sonar, P. Lawhatre, S. Ghaisas, “Automated Identification and Deconstruction of Penalty Clauses in Regulation,” *IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pp. 96-105, 2021.

[7] A.I. Antón, C. Potts. “A representational framework for scenarios of system use.” *Requirements Engineering 3* (1998): 219-241.

[8] M. Aoyama, “Persona-Scenario-Goal Methodology for User-Centered Requirements Engineering,” *15th IEEE International Requirements Engineering Conference*, pp. 185-194, 2007.

[9] R. Balebako, L. Cranor. “Improving app privacy: Nudging app developers to protect user privacy.” *IEEE Security & Privacy* 12(4): 55-58, 2014.

[10] A. Barth, A. Datta, J.C. Mitchell, H. Nissenbaum, “Privacy and Contextual Integrity: Framework and Applications,” IEEE Symp. on Sec. & Priv., 2006, pp. 184-198.

[11] R.A. Bauer. “Consumer behavior as risk taking.” *43rd National Conference of the American Marketing Assocation*, 1960.

[12] J. Bhatia, T. D. Breaux, “Towards an information type lexicon for privacy policies,” *IEEE Eighth International Workshop on Requirements Engineering and Law (RELAW)*, pp. 19-24, 2015.

[13] J. Bhatia, T. D. Breaux. “Empirical Measurement of Perceived Privacy Risk.” ACM Transactions on Computer Human Interaction, 25(6): Article 34 (December 2018), 47 pages.

[14] J. Bhatia, T.D. Breaux, J.R. Reidenberg, T.B. Norton. “A theory of vagueness and privacy risk perception.” *24th International Requirements Engineering Conference*, pp. 26-35, 2016.

[15] F. Casillo, V. Deufemia, C. Gravino, “Detecting privacy requirements from User Stories with NLP transfer learning models,” *Information and Software Technology*, v. 146, 2022.

[16] J. Cohen. “A coefficient of agreement for nominal scales,” Educational and Psychological Measurement, 20: 37-46, 1960

[17] Cronk, R. Jason. Strategic privacy by design. International Association of Privacy Professionals (IAPP), 2018.

[18] J. Dabrowski, E. Letier, A. Perini, A. Susi. “Mining User Opinions to Support Requirement Engineering: An Empirical Study.” *Advanced Information Systems Engineering (CAiSE), Lecture Notes in Computer Science,* v. 12127, pp. 401–416, 2020.

[19] Data.AI, “State of Mobile 2022”, 2022.

[20] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[21] J.L. Dupree, R. Devries, D.M. Berry, E. Lank. “Privacy Personas: Clustering Users via Attitudes and Behaviors toward Security Practices.” *CHI Conference on Human Factors in Computing Systems (CHI ’16)*, pp. 5228–5239, 2016.

[22] A. Egyed. “A scenario-driven approach to trace dependency analysis.” *IEEE Transactions on Software Engineering* 29(2): 116-132, 2003.

[23] Eurostat, “Trust, security and privacy - smartphones (2020 onwards),” 2022.

[24] J. R. Finkel, T. Grenager, C. Manning. “Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling.” 4*3nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 363-370, 2005.

[25] B. Fischhoff, P. Slovic, S. Lichtenstein, S. Read, B. Combs. “How safe is safe enough? A psychometric study of attitudes towards technological risks and benefits.” *Policy sciences*, 9: 127-152, 1978.

[26] S. Frank, A. Hakamian, L. Wagner, D. Kesim, C. Zorn, J. von Kistowski, A. van Hoorn. “Interactive Elicitation of Resilience Scenarios Based on Hazard Analysis Techniques.” *15th European Conference on Software Architecture*, 2021.

[27] E. Guzman, W. Maalej. “How do users like this feature? a fine grained sentiment analysis of app reviews.” Requirements Engineering, pp. 153–162, 2014.

[28] M. Hatamian, J. Serna, K. Rannenberg. “Revealing the unrevealed: Mining smartphone users privacy perception on app markets.” Computers & Security, 83: 332–353, 2019.

[29] J. Heaps, R. Krishnan, Y. Huang, J. Niu, R. Sandhu, R. “Access Control Policy Generation from User Stories Using Machine Learning.” *Data and Applications Security and Privacy*, 2021.

[30] G.B. Herwanto, G. Quirchmayr, A.M. Tjoa. “From User Stories to Data Flow Diagrams for Privacy Awareness: A Research Preview.” *Requirements Engineering: Foundation for Software Quality. REFSQ 2022. Lecture Notes in Computer Science*, pp. 148-155, 2022.

[31] H. Hibshi, S.T. Jones, T.D. Breaux. “A Systemic Approach for Natural Language Scenario Elicitation of Security Requirements.” *IEEE Transactions on Dependable and Secure Computing* 19(6): 3579-3591, 2021.

[32] J.-H. Hoepman, *Privacy Design Strategies (The Little Blue Book)*, self-published, 2022.

[33] M.X. Jarke, T. Bui, J.M. Carroll. “Scenario management: An interdisciplinary approach.” *Requirements Engineering* 3: 155-173, 1998.

[34] T. Johann, C. Stanik, W. Maalej. “Safe: a simple approach for feature extraction from app descriptions and app reviews.” IEEE 25th International Requirements Engineering Conference, pp. 21–30, 2017.

[35] P.N. Johnson-Laird, Philip Nicholas. *The computer and the mind: An introduction to cognitive science.* Harvard University Press, 1988.

[36] H. Kaindl, S. Kramer, R. Kacsich, “A case study of decomposing functional requirements using scenarios,” *International Symposium on Requirements Engineering*, pp. 156-163, 1998.

[37] R. Kang, S. Brown, L. Dabbish, S. Kiesler. “Privacy Attitudes of Mechanical Turk Workers and the U.S. Public.” *Symposium on Usable Privacy and Security (SOUPS)*, pp. 37-49, 2014.

[38] S. Kaplan, B.J. Garrick. “On the quantitative definition of risk.” *Risk analysis* 1(1): 11-27, 1981.

[39] M. Kassab. “An empirical study on the requirements engineering practices for agile software development.” *40th EUROMICRO Conference on Software Engineering and Advanced Applications*, pp. 254-261, 2014.

[40] L. Kof, “Scenarios: Identifying Missing Objects and Actions by Means of Computational Linguistics,” *15th IEEE International Requirements Engineering Conference*, pp. 121-130, 2007.

[41] J.R. Landis, G.G. Koch. “The measurement of observer agreement for categorical data.” *Biometrics* 1977, 33: 159-74.

[42] J.C. S. do Prado Leite, K.K. Breitman. “Experiences using scenarios to enhance traceability.” *2nd International Workshop on Traceability in Emerging Forms of Software Engineering*, p. 63-70, 2003.

[43] D. van der Linden, P. Anthonysamy, B. Nuseibeh, T. T. Tun, M. Petre, M. Levine, J. Towse, A. Rashid. “Schrödinger’s security: opening the box on app developers’ security rationale.” *ACM/IEEE 42nd International Conference on Software Engineering (ICSE ’20)*, pp. 149–160, 2020.

[44] D. Liu, A. Marcus, D. Poshyvanyk, V. Rajlich. “Feature location via information retrieval based filtering of a single scenario execution trace.” *22nd IEEE/ACM International Conference on Automated Software Engineering*, pp. 234-243, 2007.

[45] G. Lucassen, F. Dalpiaz, J.M.E.M van der Werf, S. Brinkkemper. “The use and effectiveness of user stories in practice.” *22nd International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 205-222, 2016.

[46] G. Lucassen, M. Robeer, F. Dalpiaz, J.M.E.M. van der Werf, S. Brinkkemper. “Extracting conceptual models from user stories with Visual Narrator.” Requirements Engineering 22 (2017): 339-358.

[47] S. Malmasi, A. Fang, B. Fetahu, S. Kar, O. Rokhlenko. “SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition (MultiCoNER)”, *16th International Workshop on Semantic Evaluation (SemEval)*, pages 1412-1437, 2022.

[48] A. Mavin and N. Maiden, “Determining socio-technical systems requirements: experiences with generating and walking through scenarios,” *11th IEEE International Requirements Engineering Conference*, pp. 213-222, 2003.

[49] A. McCallum, W. Li “Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced

Lexicons," 7^{th} *Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 188-191, 2003.

[50] A. H. Mhaidli, Y. Zou, F. Schaub, "We Can't Live Without Them! App Developers' Adoption of Ad Networks and Their Considerations of Consumer Risks." *15th Symposium on Usable Privacy and Security*, pp. 225-244, 2019.

[51] H. Nakayama, "seqeval: A Python framework for sequence labeling evaluation," https://github.com/chakki-works/seqeval

[52] L. Naslavsky, T.A. Alspaugh, D.J. Richardson, H. Ziv. "Using scenarios to support traceability." *3rd International Workshop on Traceability in Emerging Forms of Software Engineering*, pp. 25-30, 2005.

[53] M. Nayebi, H. Cho, G. Ruhe. "App store mining is not enough for app improvement," *Empirical Software Engineering,* (23): 2764–2794, 2018.

[54] M.C. Oetzel, S. Spiekermann. "A systematic methodology for privacy impact assessments: a design science approach." *European Journal of Information Systems*, 23(2): 126-150, 2014.

[55] A. Perrin, "Mobile Technology and Home Broadband, 2021" PEW Research Center, 2021.

[56] D. Poshyvanyk, Y-G. Guéhéneuc, A. Marcus, G. Antoniol, V. Rajlich. "Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval." *IEEE Transactions on Software Engineering* 33(6): 420-432, 2007.

[57] C. Potts, K. Takahashi, A.I. Anton. "Inquiry-based requirements analysis."" IEEE Software 11(2): 21-32, 1994.

[58] F. Pudlitz, F. Brokhausen and A. Vogelsang, "Extraction of System States from Natural Language Requirements," *IEEE 27th International Requirements Engineering Conference (RE),* 2019, pp. 211-222.

[59] J. Saldaña, *The Coding Manual for Qualitative Researchers*, Sage Pubs. 2012.

[60] N. Sannier, M. Adedjouma, M. Sabetzadeh, L. Briand, J. Dann, M. Hisette, "Legal markup generation in the large: an experience report," *IEEE 25th International Requirements Engineering Conference*, pp. 302-311, 2017.

[61] F. Schaub, R. Balebako, A.L. Durity, L.F. Cranor. "A design space for effective privacy notices." In *11th Symposium on Usable Privacy and Security*, pp. 1-17, 2015.

[62] A. Senarath, A.G.A. Nalin. "Why developers cannot embed privacy into software systems? An empirical investigation." *22nd International Conference on Evaluation and Assessment in Software Engineering*, pp. 211-216. 2018.

[63] K. Shilton, D. Greene. "Linking Platforms, Practices, and Developer Ethics: Levers for Privacy Discourse in Mobile Application Development," *Journal of Business Ethics*, 155:131–146, 2019.

[64] D. Siahaan, I.K. Raharjana, C. Fatichah. "User story extraction from natural language for requirements elicitation: Identify software-related information from online news," *Information and Software Technology*, v. 158, 2023.

[65] P. Singh Kochhar, F. Thung, N. Nagappan, T. Zimmermann, D. Lo. "Understanding the Test Automation Culture of App Developers," *IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, 2015, pp. 1-10, 2015.

[66] W.R. Shadish, T.D. Cook, and D.T. Campbell. *Experimental and Quasi-experimental Designs for Generalized Causal Inference.* Houghton-Mifflin Company, Boston, Massachusetts, 2002.

[67] A. Sleimi, M. Ceci, N. Sannier, M. Sabetzadeh, L. Briand and J. Dann, "A query system for extracting requirements-related information from legal texts," *IEEE 27th Int'l Req'ts Engr. Conf.*, 2019, pp. 319-329.

[68] A. Sleimi, N. Sannier, M. Sabetzadeh, L. Briand and J. Dann, "Automated extraction of semantic legal metadata using natural language processing," *IEEE 26th Int'l Req'ts Engr. Conf.*, 2018, pp. 124-135.

[69] P. Slovac. *The Perception of Risk (Risk, Society and Policy)*, Earthscan, 2000.

[70] C. Starr. 1969. "Social benefit versus technological risk." Science, 165, pp. 1232-1238, 1969.

[71] N. Steinfeld. "I agree to the terms and conditions: (How) do users read privacy policies online? An eye-tracking experiment," *Computers in Human Behavior*. 55: 992–1000, 2016.

[72] A. Sutcliffe. "Scenario-based requirements analysis." *Requirements Engineering Journal* 3(1): 48-65, 1998.

[73] A. Sutcliffe, "Scenario-based requirements engineering," *11th IEEE International Requirements Engineering Conference*, Tutorial, pp. 320-329, 2003.

[74] A. G. Sutcliffe, M. Ryan, "Experience with SCRAM, a SCenario Requirements Analysis Method," *Proceedings of IEEE International Symposium on Requirements Engineering*, pp. 164-171, 1998.

[75] I. Tenney, D. Das, E. Pavlick. "BERT Rediscovers the Classical NLP Pipeline," *57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, 2019.

[76] S. Thew, A. Sutcliffe. "Value-based requirements engineering: method and experience." *Requirements Eng* 23: 443–464, 2018.

[77] E. F. Tjong, K. Sang, F. De Meulder. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition." 7^{th} *Conference on Natural Language Learning at HLT-NAACL*, pp. 142–147, 2003.

[78] T. Toyama, A. Ohnishi, "Rule-based verification of scenarios with pre-conditions and post-conditions," *13th IEEE International Conference on Requirements Engineering*, pp. 319-328, 2005.

[79] A. Tversky, D. Kahneman. "Judgment under Uncertainty: Heuristics and Biases: Biases in judgments reveal some heuristics of thinking under uncertainty." *Science* 185(4157): 1124-1131, 1974.

[80] S. Uchitel, R. Chatley, J. Kramer, J. Magee. "System architecture: the context for scenario-based model synthesis." *12th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pp. 33-42, 2004.

[81] Y. Wautelet, S. Heng, M. Kolp, I. Mirbel. "Unifying and extending user story models." *26th International Conference on Advanced Information Systems Engineering*, pp. 211-225, 2014.

[82] K. Weidenhaupt, K. Pohl, M. Jarke, P. Haumer. "Scenarios in system development: current practice." IEEE Software 15(2): 34-45, 1998.

[83] A. Westin, H. Louis & ASSOCIATES. "Harris-Equifax Consumer Privacy Survey." Tech. rep., 1991. Conducted for Equifax Inc. 1,255 adults of the U.S. public.

[84] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D. Wagner, K. Beznosov. "Android Permissions Remystified: A Field Study on Contextual Integrity" pp. 499-514, 2015.

[85] J. Wittevrongel, F. Maurer. "Using UML to partially automate generation of scenario-based test drivers." *7th International Conference on Object-Oriented Information Systems*, pp. 303-306, 2001.

[86] A. Woodruff, V. Pihur, S. Consolvo, L. Schmidt, L. Brandimarte, A. Acquisti. "Would a privacy fundamentalist sell their DNA for $1000... if nothing bad happened as a result? The Westin categories, behavioral intentions, and consequences." *Symposium on Usable Privacy and Security (SOUPS)*, pp. 1-18, 2014.

[87] R.K. Yin. *Case study research*, 3rd ed. In Applied Social Research Methods Series, v.5. Sage Publications, 2003.

[88] L. Zhuang, L. Wayne, S. Ya, Z. Jun. "A Robustly Optimized BERT Pre-training Approach with Post-training," *20th Chinese National Conference on Computational Linguistics*, pp. 1218–1227, 2021.