Scrutinizing Privacy Policy Compliance of Virtual Personal Assistant Apps

Fuman Xie
University of Queensland
Australia

Yanjun Zhang
Deakin University
Australia

Chuan Yan
University of Queensland
Australia

Suwan Li
Nanjing University
China

Lei Bu
Nanjing University
China

Kai Chen
Chinese Academy of Sciences
China

Zi Huang
University of Queensland
Australia

Guangdong Bai*
University of Queensland
Australia

ABSTRACT

A large number of functionality-rich and easily accessible applications have become popular among various virtual personal assistant (VPA) services such as Amazon Alexa. VPA applications (or *VPA apps* for short) are accompanied by a *privacy policy document* that informs users of their data handling practices. These documents are usually lengthy and complex for users to comprehend, and developers may intentionally or unintentionally fail to comply with them. In this work, we conduct the first systematic study on the privacy policy compliance issue of VPA apps. We develop Skipper, which targets Amazon Alexa skills. It automatically depicts the skill into the *declared privacy profile* by analyzing their privacy policy documents with Natural Language Processing (NLP) and machine learning techniques, and derives the *behavioral privacy profile* of the skill through a black-box testing. We conduct a large-scale analysis on all skills listed on Alexa store, and find that a large number of skills suffer from the privacy policy noncompliance issues.

nlp+机器学习（PP）黑盒测试（VPA）

CCS CONCEPTS

• **Security and privacy → Web application security**.

KEYWORDS

Virtual Personal Assistant, privacy compliance, Alexa skills

ACM Reference Format:
Fuman Xie, Yanjun Zhang, Chuan Yan, Suwan Li, Lei Bu, Kai Chen, Zi Huang, and Guangdong Bai. 2022. Scrutinizing Privacy Policy Compliance of Virtual Personal Assistant Apps. In *37th IEEE/ACM International Conference on Automated Software Engineering (ASE '22), October 10–14, 2022, Rochester, MI, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3551349.3560416

*Corresponding author.

ASE '22, October 10–14, 2022, Rochester, MI, USA

ACM ISBN 978-1-4503-9475-8/22/10…$15.00
https://doi.org/10.1145/3551349.3560416

1 INTRODUCTION

The flourishing of Internet of Things (IoT) in recent years has made massive interactive smart devices an integral part of our daily lives. This brings about a revolution in human computer interaction (HCI). The graphics-based HCI is reshaped towards the pure voice-based interaction to facilitate user engagement anytime and anywhere. Various AI (artificial intelligence)-backed virtual personal assistant (VPA) services, e.g., Amazon Alexa and Google Assistant, are easily accessible on smart devices. The users could give verbal commands (or *utterances*) to invoke a broad spectrum of functions, such as making phone calls and controlling smart devices. As reported by Statista [17], VPA services have been available on billions of devices around the world.

Inspired by the great success of mobile app ecosystem, the VPA services enable third-party developers to create VPA apps (e.g., *skills* in Amazon Alexa and *actions* in Google Assistant) to enrich their capabilities. The developers release their apps through app stores, which makes the apps easily accessible to the user by simply giving the smart speaker an utterance like "*Alexa, open* <app name>". The openness of this model, however, renders it possible for dishonest apps to collect the user's personal information, such as name, location, gender and age. Although the VPA services enforce a *permission-declaring* mechanism [21, 27] where the app must request appropriate permissions to access sensitive information, this does not prevent the app from gathering data from their conversations with the user, i.e., the notorious *runtime information gathering* (RIG) threat [39, 48].

Personal data protection has gained a great deal of attention in recent years. Many countries have enacted legislations to regulate the collection, use and sharing of personal data, such as the European Union (EU) General Data Protection Regulation (GDPR) [5] and California Consumer Privacy Act (CCPA) [11]. These regulations impose strict obligations on data controllers and data processors, where any infringement of user privacy could lead to large penalty. In response to these increasingly stringent regulations worldwide, VPA service providers have taken steps to enforce privacy features. Skill developers are now recommended to release privacy policies to disclose the skills' user data handling practices.

Security implications and privacy concerns of the VPA ecosystem have also attracted high attention from the research community.

On the security of VPA apps, efforts [29, 31, 47, 49] have been made to detect the *squatting attack* where an attack app is created with an invocation name that sounds similar to a legitimate app (e.g., "full moon" v.s. "four moon") to impersonate the latter. On their privacy properties, existing studies focus on detecting and defending against personal information querying [28, 42]. These however offer only a preliminary view of the RIG threat. Querying personal information is not necessarily malicious, as the app may need particular information to provide relevant service. For example, collecting location by a weather forecast app should be considered benign, as long as it discloses its data handling practices and properly follows its privacy policy. Therefore, the complementary problem of *whether VPA apps are developed to comply with their privacy policy and how to systematically scrutinize this compliance* becomes critical.

Our Work. In this work, we explore this problem in the VPA app (i.e., *skill*) ecosystem of Amazon Alexa, the most popular VPA service. We develop Skipper (skill privacy policy examiner), which adopts a three-phrase workflow. The first phase aims to derive the skill's *declared privacy profile* from the data protection practices it describes in the privacy policy document. The challenge to overcome is to infer context-sensitive and fine-grained information from the documents that are written in various styles. Skipper trains a *paragraph-level* classifier to split the privacy policy document into sections according to the policies they describe, e.g., policies on types of collected data and policies on child protection mechanisms, and then uses *sentence-level* Natural Language Processing (NLP) techniques to extract information specific to each category, e.g., the concrete data types the skill collects.

In the second phase, Skipper creates privacy-specific test cases to check the skill's runtime data handling behaviors. The unique challenge is to trigger the data handling behaviors of the skill, which is a pure black box that takes utterances in natural language as the only way for interaction. Skipper implements its testing based on a chat-box based skill tester from a recent study [28] and its enhancement [33] with functionality-relevant utterances to drive the skill to its main user interface, from which Skipper starts feeding its test cases to trigger the behaviors of interest. By processing the collected log that includes the skill's (active) queries and (passive) responses, the third phase of Skipper depicts the skill's *behavioral privacy profile* and checks it against the declared profile for any noncompliance.

We conduct a large-scale study on all 61,505 Amazon Alexa skills to understand the *status quo* of the privacy policy compliance in existing real-world VPA apps. Skipper tests more than 20,000 skills in total, including 5,024 skills that have released valid privacy policies. It reveals that the current status of privacy policy compliance of skills is worrying. Overall 1,012 noncompliance issues are found, and most of them lie in the *under-declaration* of data collection behaviors. Some skills are found having noncompliance issues in children's data protection, region-specific policy and data retention.

Contributions. The main contributions of this work are as follows.

- **Understanding the privacy policy noncompliance issue on a large scale.** We conduct the first comprehensive study on the privacy policy compliance of skills. Our work detects inconsistencies between skills' runtime behaviors and the statements in their privacy policies disclosed to users.

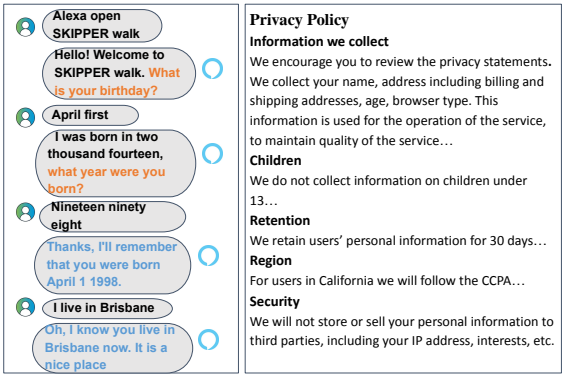

Figure 1: A running example: a skill that collects user data

- **A systematic assessment approach.** We propose a series of analysis techniques to automatically identify privacy noncompliance issues from skills. Our approach features the profile inference based on sentence-level privacy policy interpretation, and the black-box testing guided by the declared privacy profile. It is thus more targeted than general-purpose skill testers based on chatbots.
- **Revealing the *status quo* of privacy policy compliance of VPA apps.** We present the landscape of privacy policy compliance of Alexa skills, the apps of the most popular VPA service. Our findings reveal that the current privacy policy compliance remains problematic. Our work should raise an alert to users, and encourage VPA service provider to put in place regulations on the privacy policy compliance.

2 A RUNNING EXAMPLE AND DEFINITIONS

In this section, we use a running example shown in Figure 1 to introduce skills and the privacy policy noncompliance issue (Section 2.1), and then present the problem definition (Section 2.2).

2.1 A Running Example and Motivation

Alexa Skills. Alexa adopts a development ecosystem that is similar to the one of mobile apps. Any third-party developer can upload a skill to the Alexa skill store, and also provide the skill's (supposedly unique) innovation name, one-sentence description, detailed description, icon, category and example phases that guide users how to interact with it. After passing a validation testing by Amazon, the skill becomes accessible to users. The user can start it by simply saying an utterance "*Alexa open + <skill's invocation name>*" (e.g., the first utterance in our running example) to the smart speaker. Afterwards, the user and the skill can have verbal conversations.

Privacy Policies. During conversations with the user, the skill can collect the user's personal data by asking specific questions, e.g., the first and second sentences from the skill in our running example, or by recording the user's utterances, e.g., the fourth utterance from the user in our running example. This not only causes privacy concerns of the user, but also put the skill developer at risk of huge penalties from the data protection regulations such as GDPR. As a result, Amazon starts requiring the developers to release a privacy policy document to disclose its data handling practices [10]. Skills must ensure that "*collection and use of that* (personal) *information complies with your* (skills') *privacy notice and*

Table 1: Eight types of policies in the privacy policy document

	Category	Description	Example
PP-1	TYPES	Types of personal data collected by the skill	*"We may collect your name, birth date, ..., and any other information you may voluntarily provide to us."*
PP-2	CHILDREN	Privacy policies for child protection mechanisms	*"We are in compliance with the requirements of COPPA. We do not collect any information from anyone under 13 years of age. Our services are all directed to people who are at least 13 years old or older."*
PP-3	REGIONS	Data protection mechanisms for special regions	*"California residents, called "consumers" in the CCPA, have the following rights: ... the right to request that we delete any of your personal information that we collect from you and retained ..."*
PP-4	RETENTION	How long the skill keeps the collected data	*"We will retain User Provided data for as long as you use Bathroom Sidekick and for a reasonable time thereafter." "We will retain Automatically Collected information for up to 1 month."*
PP-5	SECURITY	How the skill protects collected data	*"... these security measures include password protected directories and databases to safeguard your information, SSL (Secure Sockets Layered) technology to ensure that your information is fully encrypted and sent across the Internet securely or PCI Scanning to actively protect our servers from hackers and other vulnerabilities. "*
PP-6	USER_RIGHT	The rights that the user (i.e., the data owner) has	*"... if you have signed-up to receive our email marketing communications, you can unsubscribe any time by clicking the "unsubscribe" link included at the bottom of our emails."*
PP-7	UPDATE	Whether the privacy policy will be updated	*"Please note that this Privacy Policy may be periodically updated. Please refer to our website for the latest Privacy Policy that is in force. "*
PP-8	DATA_USE	How the skill will use personal data	*"The information we collect is used to improve our website in order to better serve you, to allow us to better service you in responding to your customer service requests, ..., and to quickly process your transactions."*

Table 2: Definition of [*Category Vector*]

Type	Structure of [*Category Vector*]	Description
TYPES	[name, email address, phone number, postcode, ...]†	A dynamic object vector. A data type is included in [*Category Vector*] when the skill states collecting the type of data from the user.
CHILDREN	[age, [*CTypes*] : TYPES.[*Category Vector*]]	It includes the age boundary that the skill uses to recognize children, and the types of data collected from child users.
REGIONS	[region, *deletion*]	It includes the region where the skill has special policies, and whether the user can request for data deletion (see Section 4.2.2 for details).
RETENTION	[period]	It includes the retention period. If the skill does not store data, the period is 0.

† The full list of data types learned by Skipper (detailed in Section 4.2) are listed below.

All data types learned by Skipper	"name", "email address", "phone number", "billing address", "birth date", "age", "user id", "gender", "location", "job title", "phonebook", "sms", "income", "ip", "internet protocol", "marital", "social security number", "credit card", "type of browser" ,"browser version", "operate system", "postal address", "shipping address", "postcode", "profile", "education", "occupation", "student", "software", "driver", "insurance", "health", "signature", "province", "time zone", "isp", "tax", "device id", "domain name", "prior usage", "cookie", "web page", "interact site", "device information", "dash cam", "log data", "page service visit", "time spend page", "time date visit", "time date use service", "demographic information", "country", "usage pattern", "language", "reminder", "alexa notification", "amazon pay"

all applicable laws" [23]. By referring to literature on privacy policy studies [36, 37] and reviewing privacy policy documents of major developers like Google and Amazon, we summarize eight categories of privacy policies that are commonly included in privacy policy documents, as listed in Table 1.

2.2 Problem Definition

The core idea of Skipper is to portray the skill as its *declared privacy profile* and *behavioral privacy profile*, and then to check these two profiles for inconsistencies. Among the eight types of policies (see Table 1), we include those *testable* or *partially testable ones* in the profiles, including **PP-1** (TYPES): policies on the types of data to collect, **PP-2** (CHILDREN): policies on child protection mechanisms, **PP-3** (REGIONS): policies on protection mechanisms for special regions, and **PP-4** (RETENTION): policies on data retention period. The remaining four types of policies either require extra information (***PP-7*** UPDATE) and full manual efforts (***PP-6*** USER_RIGHT) to check, or involve server-side behaviors (***PP-5*** SECURITY and ***PP-8*** DATA_USE), so we exclude them from Skipper's scope.

Definition 1. [**Privacy Profile**] A privacy profile of a skill is a three-tuple (Category, *covered*, [*Category Vector*]), where Category indicates the policy type (**PP-1** to **PP-4**), *covered* indicates whether the policy type of Category is covered by the profile, and [*Category Vector*] is an object specific to Category. The data structure of [*Category Vector*] for each category is defined in Table 2.

We use $\mathcal{D}$ to indicate the declared privacy profile of a skill and $\mathcal{B}$ the behavioral privacy profile. For simplicity, we abuse $\mathcal{D}/\mathcal{B}$.Category to refer to [*Category Vector*] of Category, e.g., $\mathcal{D}$.CHILDREN instead of $\mathcal{D}$.[*Category Vector*] *s.t.* $\mathcal{D}$.Category=CHILDREN.

Definition 2. [**Compliance**] A skill's behavior complies with its privacy policy if the following four rules hold.

[R1] $\mathcal{B}$.TYPES $\subseteq$ $\mathcal{D}$.TYPES,

[R2] $\mathcal{B}$.CHILDREN.[*CTypes*] $\subseteq$ $\mathcal{D}$.CHILDREN.[*CTypes*] when $\mathcal{B}$.CHILDREN.*age* $\leq$ $\mathcal{D}$.CHILDREN.*age*,

[R3] The skill allows data deletion when $\mathcal{D}$.REGIONS.*region* = California, and

[R4] $\mathcal{B}$.RETENTION.*period* $\leq$ $\mathcal{D}$.RETENTION.*period*.

Below we explain these four rules and describe the violations that are considered as noncompliance issues in this work.

[V1] ***Under-claim of personal information collection*** (violation of R1). An honest skill should declare all types of personal data it collects. Therefore, Skipper raises an *under-claim alert* when the skill collects certain types without disclosing them in its privacy policy, i.e., $\mathcal{B}$.TYPES $-$ $\mathcal{D}$.TYPES $\neq \emptyset$. We note that $\mathcal{B}$.TYPES = $\mathcal{D}$.TYPES is not enforced by Skipper as the skill may conduct data collection through other channels, e.g., Alexa APIs [16].

[V2] ***Misconduct on children's data*** (violation of R2). Different countries have different definitions of the age boundary of children. For example, the US and Canada take users under the age of 13 as children, while the UK and Germany take 16 as the boundary [22]. Therefore, Skipper conservatively sets $\mathcal{D}$.CHILDREN.*age* as 13 if no age boundary is defined or mentioned in the privacy policy, and raises an alert if $\mathcal{B}$.CHILDREN.[*CTypes*] $-$ $\mathcal{D}$.CHILDREN.[*CTypes*] $\neq \emptyset$ when $\mathcal{B}$.CHILDREN.*age* $<$ $\mathcal{D}$.CHILDREN.*age*.

[V3] ***Failure to delete personal data upon request*** (violation of R3). Skipper checks REGIONS by a region parameter extracted from priacy policies. Among current regional regulations, GDPR

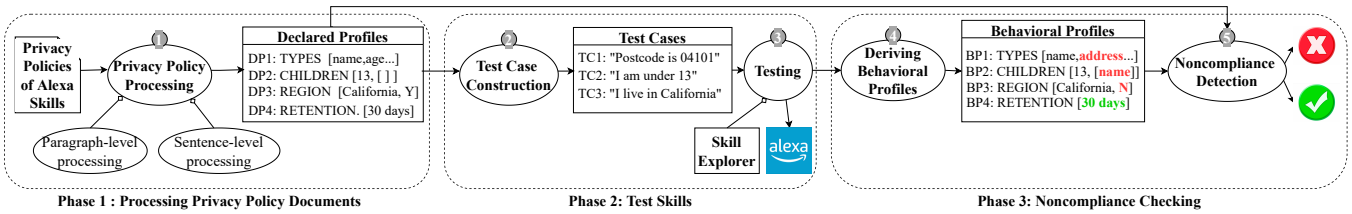

Figure 2: Workflow of Skipper

is pervasively adopted among existing skills. Only a few explicitly define their regional policies, and almost all of them refer to the California Consumer Privacy Act (CCPA). Since CCPA specifically states that California residents have the right to make data deletion requests [20], Skipper focuses on checking whether the deletion request is properly handled by the skill. Our approach can be extended to support other regional privacy policies like the recent Colorado Privacy Act (CPA) [9], if any skill is found adopting them.

[V4] *Over-retention* (violation of R4). This occurs when the skill retains collected data longer than its declared retention period ($\mathcal{B}$.RETENTION.*period* > $\mathcal{D}$.RETENTION.*period*). For skills that do not provide any RETENTION policy (including those not providing privacy policy), Skipper sets their $\mathcal{D}$.RETENTION.*period* = 0.

Assumption on intention of data collection. Skipper assumes that all queries for personal data are for the purpose of data collection. We do not aim to recognize other purposes, although there may be queries for benign use. For example, the query "*what year were your born*" in our running example (Figure 1) may be used to determine whether the user is a child.

3 OVERVIEW OF SKIPPER

We design Skipper as a three-phase approach that consists of *processing privacy policy documents*, *testing skills* and *checking for privacy policy noncompliance*, as shown in Figure 2.

Phase 1: Processing Privacy Policy Documents. This phase aims to extract the skill's declared privacy profile (i.e., the $\mathcal{D}$), given its privacy policy document written in natural language as input. There is still lack of a privacy policy analyzer in the literature that is capable of inferring all information needed to construct the complete declared profile. Skipper builds a fine-grained analyzer for constructing [*Category Vector*], using machine learning and NLP techniques. This phase is detailed in Section 4.

Phase 2: Testing Skills. This phase aims to trigger and capture the skill's runtime behaviors in user data handling. It consists of two main steps, i.e., test case generation and execution. During the test case generation, Skipper constructs test cases that are specific to the Category and the [*Category Vector*]. As skills take the utterances in natural language as inputs, they are tolerant to the format of the utterances. This saves Skipper from creating a massive number of mutants. Skipper uses an enhancement [33] of the state-of-the-art skill tester SkillExplorer [28] to drive the execution of the skill. It feeds its test cases when the skill is in the main user interface, so as to trigger more behaviors on data collection and handling. This phase is detailed in Section 5.

Phase 3: Noncompliance Checking. This phase aims to derive the skill's behavioral privacy profile (i.e., the $\mathcal{B}$) and detects privacy policy noncompliance issues by analyzing the collected test logs which consist of the conversations with the skill. Skipper makes decision based on the skill's queries and responses to our test cases. This phase is detailed in Section 6.

Privacy Policy Documents Collection. Since there is a lack of privacy policy dataset available for our analysis, we first create one by ourselves. We refer to a latest skill dataset named UQ-AAS21 [44] for a complete list of existing skills. It contains 65,195 skills on the Alexa app store up to July 2021. We build a crawler to retrieve the privacy policy document of each skill with the link obtained from the dataset, and have obtained in total 7,513 privacy policy documents. We then filter the obtained documents to ensure their quality. We remove those whose size is smaller than 2KB or number of sentences is fewer than 20, following a recent study [35] that handles Android app privacy policy documents. Policies that are not written in English are deleted, based on the language recognition of a tool named langdetect [15]. We also exclude those documents that do not include the keyword "*privacy*" or "*user information*". After the filtering, 5,829 privacy policies are kept. This suggests that the availability and quality of the privacy policies of existing skills are terribly unsatisfactory, which is to our great surprise, given that many major companies have undergone huge penalty due to data protection violations, such as Amazon [8] and Google [6]. This further motivates our efforts in checking the privacy weaknesses in these VPA apps.

4 DETERMINING DECLARED PROFILES

Some existing studies have proposed document-level [36, 50] and paragraph-level [40, 43] processing of privacy policy document, when they check whether a given privacy policy has covered the GDPR principals. However, such high-level techniques are too coarse-grained to be applicable by Skipper. On the other hand, using only sentence-level processing [34, 41] may lead to false positives or negatives, as a sentence appearing in different sections of the privacy policy document may have different implications. For example, developers usually refer to the way they protect user data in a *security* section, as shown in Figure 1. The statement "*we will not store or sell your personal information to third parties, including your IP address, interests, etc.*" could be recognized as a sentence of TYPES by an analyzer that does not consider the context.

To address this, we propose a two-step method that breaks down the document into paragraphs (Section 4.1) and then further into sentences (Section 4.2), so that NLP techniques can extract accurate information. This is inspired by studies [19, 32] that analyze policies and documents of software such as Android apps. In this section, we detail our methods for the entire process.

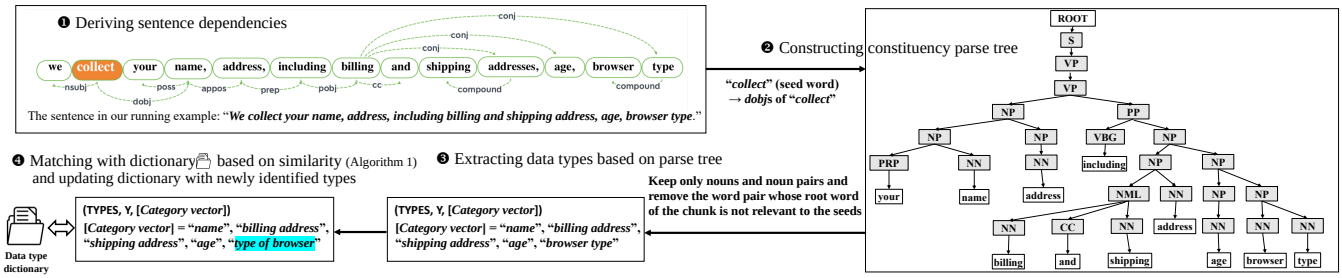

Figure 3: The sentence-level processing (using TYPES as an example)

Table 3: Coverage of each privacy policy category.

F: Frequency, C: Coverage

Type	Examples of headings	F	C
TYPES	"*Personal Information We Collect*", "*Types of Data Collected*".	4,462	77%
CHILDREN	"*children privacy*", "*Our Policy Towards Children*", "*Children Under Thirteen*	1,643	33%
REGIONS	"'*Notice for California Residents*", "*Additional Information for California Residents*"	1,299	23%
RETENTION	"*How long we can hold your data for*", "*Data retention period*"	1,000	17%

4.1 Paragraph-level processing

We first train a classifier to categorize paragraphs according to the policies they describe. We observe that the vast majority of documents (>91%) follow the $(\langle \texttt{Heading} \rangle \langle \texttt{Paragraph} \rangle^{+})^{+}$ format. The heading well explains the purpose of the paragraphs following it, and to obtain the headings is feasible, as they are typically highlighted with HTML tags (e.g., <h1> and <b>) or enumeration. We thus use the headings to train the classifier, and the paragraphs led by a heading are categorized into the heading's class.

Data labeling. We randomly select 60 from the 5,829 collected documents, and manually label each of their headings (591 in total) after reading the paragraphs, with nine labels (i.e., **PP-1** to **PP-8** and NOT_INTERESTED). In the labelling process, two authors (one with law expertise) independently conduct the labelling. They reach consistent labels in 98% of all cases. For any disagreement, another author is involved to resolve the conflict.

Training. After labeling, we find that the headings usually contain a few feature words that can be used as identifiers. For example, the paragraphs with a heading containing "*collect*" are likely to describe data types (i.e., TYPES); the paragraphs with a heading containing "*children*" or "*parent*" are likely about specific policies for children (i.e., CHILDREN). In light of this, we train a Bayesian multi-label classifier, which is known to have strong capability of recognizing such patterns and handling multiple labels [38]. We randomly divide the dataset into 80% and 20% (i.e., the hold-out cross-validation). The former is used as training data, and the latter is used for testing. The trained classifier reaches a classification accuracy of 0.85.

We then apply it to the rest of the documents. In Table 3 (column 3 and 4), we summarize the classification results. The frequency and coverage indicate the count and percentage of the documents that include the corresponding headings. We can see that among 5,829 privacy policies, most (77%) have declared the type of user data to collect. They generally place less emphasis on child policies (33%), regional policies (23%) and data retention (14%) though.

Validity of heading-based classification. Among all documents, Skipper reports that 562 have no headings. To confirm this, we conduct a manual review on them. We find that most of them are badly formatted or poorly readable. They may have headings, but the headings are not highlighted with HTML tags or enumeration. In addition, some put all sentences in one paragraph, and some have no punctuation used at all throughout the whole document. For such cases, we attempt to manually split them according to the $(\langle \texttt{Heading} \rangle \langle \texttt{Paragraph} \rangle^{+})^{+}$ format, and this manages to label another 142 documents. The remaining documents neither include relevant contents nor are readable, so we consider that they fail to provide valid privacy policy documents.

4.2 Sentence-level processing

After obtaining the paragraphs of each policy type, Skipper extracts information needed for constructing the privacy profile from their sentences. In this section, we detail the proposed techniques.

4.2.1 TYPES. We extract the TYPES paragraphs from the 60 documents, and then manually label the sentences into two categories (Y/N) according to their relevance to concrete data types. The labelling for our running example is shown below.

> We encourage you to review the privacy statements. [N] We collect your name, address, including billing and shipping addresses, age, browser type. [Y] This information is used for the operation of the service, to maintain quality of the service, and to provide general statistics regarding use of the our website. [N]

The labeling process is the same as in paragraph labelling (see Section 4.1). After it, we obtain a dataset with 499 labelled sentences. We use them to train another Bayesian classifier. It reaches a high accuracy of 0.98, as the sentences have a unique pattern of a special verb (e.g., "*collect*") followed by multiple nouns.

With the trained classifier, Skipper can identify the sentences that describe the types of collected data. After obtaining such a sentence, it uses a *domain-adapted dependency and constituency parsing* to understand its structure and determine the [*Category Vector*]. We use our running example to illustrate this process in Figure 3. Skipper first uses Spacy [14] to construct a dependency-based parse tree, which enables it to infer semantics of the sentence based on the grammatical relationships among the words (step ❶ and ❷ in Figure 3). Considering that the plain parse trees are less informative with respect to our problem domain, we update the tree by annotating a few keywords of the problem domain. In particular, we define a list of seed words that are the synonym of *obtain*, including *access, check, collect, gather, know, receive, save,* and *store*. Skipper scans the sentence for the word that has the same POS (part-of-speech) token attribute with the seed words, and checks whether it is a synonym of any seed word. If so, Skipper extracts its

Algorithm 1 Phrase Similarity

```
   Input: phrases p1, p2
          include WordNet.path_similarity as W.Sim
   Output: similarity of p1 and p2
 1: function Phase_sim(p1, p2)
 2:     sim ← 0
 3:     if Word_count(p1)=1 and Word_count(p2)=1 then
 4:         sim ← W.Sim(p1, p2)
 5:     else
 6:         for each word μ in p1 do
 7:             wset_μ ← Find_3_synonyms(μ) ∪ {μ}
 8:             for each word ν in p2 do
 9:                 wset_ν ← Find_3_synonyms(ν) ∪ {ν}
10:                 sim ← sim + Max_sim(wset_μ, wset_ν)
11:             end for
12:         end for
13:         sim ← sim/min(Word_count(p1), Word_count(p2))
14:         return sim
15:     end if
16: end function
17: function Max_sim(wset1, wset2)
18:     sim ← 0
19:     for each pair (ϕ, ψ) in wset1 × wset2 do
20:         if W.Sim(ϕ, ψ) > sim then
21:             sim ← W.Sim(ϕ, ψ)
22:         end if
23:     end for
24:     return sim
25: end function
```

corresponding dependents in the tree. In this way, Skipper obtains the phrases/areas in the sentence that may contain the objects related to data types (step ❸).

Now that Skipper can extract data types from a sentence, it starts constructing a dictionary which includes all types of personal data that skills may collect, to address the challenge that there is no existing domain-specific dictionary available for our study. A naive way is to add all noun chunks extracted from the sentences into the dictionary after cleaning and stemming. However, some phrases express the same meaning but use different words, for example, "*geographic information*" and "*location data*". We thus propose an algorithm to merge such phrases based on similarity, as shown in Algorithm 1. When the given two phrases are both words, Skipper uses the `path_similarity` function of WordNet [18] to compute their similarity (line 3&4 of Algorithm 1). When either of them includes more than one word, Skipper finds three synonyms of each word from the CORPUS package of NLTK [7] (line 6-9). It groups the original word and its synonyms as a *word set*, and then the similarity of the most two similar words in two word sets is used to represent the similarity of the two word sets (Max_sim in line 17-25). The similarity of two phrases is calculated through the similarity of the word sets of each pair of words μ and ν in the two phrases (line 10&13). When a phrase pair has a similarity score >0.8 (empirically set), Skipper treats them as a single phrase.

Applying the proposed techniques of identifying and processing relevant sentences from all privacy policy documents, Skipper obtains a dictionary of all data types as listed at the bottom of Table 2. After that, Skipper can conduct a key phrase-based pattern matching [40, 43, 46, 50] with the vocabularies in the dictionary to identify $\mathcal{D}$.TYPES for each skill (step ❹).

4.2.2 `CHILDREN`, `REGIONS` *and* `RETENTION`. For these three categories, Skipper follows the same methodology for `TYPES` in identifying relevant sentences. It has subtle differences in processing these sentences.

For the `CHILDREN` category, Skipper first extracts numbers in the sentences based on the `POS` token attribute (i.e., the `NUM` tokens) labelled by Spacy. It can recognize a number in any format, such as "13" and "thirteen". A number less than 20 is taken as the age boundary based on which the skill decides a user as a child. If the skill mentions that it follows the Children's Online Privacy Protection Rule (COPPA), Skipper uses 13 as the age boundary, as COPPA imposes special restrictions for "*children under 13 years of age*" [1]. For those documents that do not explicitly define the age period of children, Skipper follows Alexa's official documentation [22] on the age boundary. It summarizes that either the age of 13 (e.g., the US and Canada) or 16 (e.g., the UK and Germany) is taken as the age boundary by the regulations of most countries. Therefore, Skipper conservatively sets $\mathcal{D}$.`CHILDREN`.*age* as 13 for such skills.

When Skipper recognizes `TYPES` sentences in `CHILDREN` paragraphs, it takes the data types extracted from them as what the skill collects from child users (i.e., $\mathcal{D}$.`CHILDREN`.[*CTypes*]). When a skill says it does not collect any data from children, Skipper sets $\mathcal{D}$.`CHILDREN`.[*CTypes*] as $\emptyset$.

For the `RETENTION` category, Skipper recognizes numbers and checks whether they are followed by a time-related word, such as "hour", "day", "month" and "year". Then this time period is taken as the data retention period of the skill. For `REGIONS` category, Skipper focuses on checking CCPA region policy, which explicitly states that California residents have the right to make data deletion requests [20]. Skipper determines that the skill allows deletion requests if it detects non-negative sentences containing keywords like "delete" and "erase".

5 TESTING SKILLS

As the code bases of skills are unavailable to the analyst, we take use of black-box testing to understand their behaviors. This involves generating test cases (Section 5.1) and feeding them into the skills through their VUIs (Section 5.2).

5.1 Test Case Construction

5.1.1 `TYPES` *Test Cases.* Among the full list of personal data types obtained by Skipper (see Table 2), some are irrelevant to VPA apps, e.g., *type of browser*, *browser version* and *web pages*. They are included by some developers may be because they reuse privacy policy documents for web applications. Some data types are not testable because Skipper has no access to the network traffic and skills' server-side behaviors, e.g., *cookies* and *usage pattern*. Therefore, we cover all testable types in Alexa's permission list[1]. In addition, Skipper includes *date of birth* and *age* data types to test `CHILDREN` policies.

This results in seven types of sensitive personal data for Skipper to test, including *name*, *date of birth*, *age*, *location*, *phone number*, *email*, and *postcode*. Skipper's list covers the data types that are most extensively collected by dishonest skills shown in existing

[1]The permissions available for custom skills include device address, customer name, customer email address, customer phone number, lists write, lists read, Amazon Pay, reminders, location services, Skills personalization [21].

Table 4: Test cases for TYPES †

Code	TYPES Test Case	Data Type
TC101	"*Brisbane*", "*I live in Brisbane*"	Location
TC102	"*Postcode is 04101*", "*04101*"	Postcode
TC103	"*My birthday is 25th of December*"	Birthday
TC104	"*My email is xxx@gmail.com*"	Email
TC105	"*My phone number is 0450419999*", "*0450419999*"	Phone Number
TC106	"*My name is James Smith*", "*James Smith*"	Name
TC107	"*I am 20 years old*"	Age

Table 5: Test Cases for CHILDREN

Code	CHILDREN Test Case	Combination with TYPES Test Case
TC201	"*I am under 13/16/18/*$\mathcal{D}$.CHILDREN.age"	plus TC101 to TC106
TC202	"*I am 10/14/17/*$\mathcal{D}$.CHILDREN.$age - 1$ *years old*"	plus TC101 to TC106
TC203	"*Ten/Fourteen/Seventeen years old*"	plus TC101 to TC106
TC204	"*I am a child/teenager*"	plus TC101 to TC106
TC205	"*I am a kid*"	plus TC101 to TC106
TC206	"*I was born on April 1th, 2010/2006*"	plus TC101 to TC102, TC105 to TC106
TC207	"*April 1th, 2010/2006*"	plus TC101 to TC102, TC105 to TC106

studies [26, 28], and Skipper's testing could be extended to include other types. For each of the data types, we design one or two test cases, as listed in Table 4.

5.1.2 CHILDREN *Test Cases.* We consider two types of test cases under this category. The first type includes utterances. As listed in Table 5, the test cases aim to make the skill to recognize the user as a child during the conversation. The column 2 of Table 5 presents the corresponding test cases. They are mostly defined around the possible age boundaries. Besides 13, 16 and $\mathcal{D}$.CHILDREN.*age* (see Section 4.2.2), Skipper also designs test cases around 18, as a few skills take 18 as the age boundary of children. The CHILDREN test cases will be combined with TYPES test cases (column 3) to check the skill's behaviors of collecting children's personal data. As the CHILDREN test cases contain some data types that may contradict TYPES, e.g., "*age*" and "*date of birth*" (TC206, TC207), we exclude the corresponding TYPES test cases in the combination.

Second, we create two Amazon accounts of Amazon's official non-adult profiles [4], a child profile (under the age of 13) and a teens profile (between 13 to 17). Skipper then uses them to interact with the skill. It feeds the skill with TYPES test cases only (column 3 of Table 5) to check whether the skill collects children's data.

5.1.3 REGIONS *Test Cases.* Skipper first designs utterances from {"*I*"} × {"*live in*", "*am resident of*", "*am from*"} × {"*California*", "*Los Angeles*"} to make the skill recognize "the user" as a California resident. It then combines with TYPES test cases to trigger the skill's behaviors in recording personal data. After that, it requests the skill to delete the collected information with {"*delete*", "*erase*", "*remove*"} × {"*my information*", "*personal information*", "*my data*"}. Besides the utterance test cases, we also create a user profile in Amazon with a fake address in California[2] and use a California VPN to interact with skills. With this profile, Skipper skips the utterances that inform the skill of its region, and feeds the skill with the TYPES test cases and deletion requests.

5.1.4 RETENTION *Test Cases.* As the RETENTION behaviors are temporal, Skipper creates no new utterances but reuses the TYPES test cases. It first runs them with a clean user profile (i.e., to simulate a user who never uses any skills). After declared retention period (i.e., $\mathcal{D}$.RETENTION.*period*), Skipper restarts the skill and interacts with it without giving any personal data. It then checks whether the skill still remembers the TYPES data (to be detailed in Section 6.2). Due to the time limitation, we test only those skills that have a retention period shorter than 30 days.

5.2 Skill Testing

After generating test cases, Skipper proceeds with testing the skills.

Scope. For TYPES category, according to GDPR Article 13 [3], the data controller is mandated to disclose the data they collect from users. Therefore, Skipper attempts to test every skill for TYPES compliance no matter it provides a privacy policy document or not. For CHILDREN and REGIONS categories, Skipper only tests those skills which provide CHILDREN and REGIONS sections in their privacy policy documents. For RETENTION category, Skipper tests all skills that it finds with data collection behaviors during TYPES testing.

Skill execution. The challenges to be addressed by Skipper are twofold. First, Skipper should trigger the skill's behaviors as many as possible, so that the extracted behavioral profile can be complete. Second, Skipper should feed the designed test cases in a suitable context, so that relevant behaviors can be captured.

Skipper relies on a state-of-the-art VPA app tester called SkillExplorer [28] and its enhancement called VITAS [33] to implement our skill testing. SkillExplorer is a chatbot-like tester that is specifically designed for interacting with skills based on NLP techniques to understand queries from skills and generate corresponding utterances. It interacts with skills through Alexa Simulator [24] and obtains their responses in text. VITAS enhances it with two types of new utterances. The first type includes the verb+noun phrases extracted from the skill documents. The other type of utterances are derived from the responses of the skill upon a "*help*" utterance, e.g., "*you can say a location to check the weather*".

Skipper benefits from the following two main capabilities of the testers. First, their capability of answering the skill's queries based on NLP techniques and maintaining long conversations with the skill enables Skipper to drive the skill to a deep state. Second, Skipper reuses their interaction module with the Alexa Simulator to feed its test cases. In particular, Skipper invokes SkillExplorer to run the skill three times. In each execution, SkillExplorer interacts with the skill till the skill ends. As the testing process is a stochastic process (due to the chatbot based test case generation and the random test case selection), the longest dialog is likely to have led the skill to a deep site of its VUI. Therefore, Skipper feeds the prefix of the longest dialog to drive the skill to an intermediate state, from which it feeds the test cases. As most test cases may be rejected by the skill, Skipper restarts the skill after collecting its responses. It repeats this process until all test cases have been fed into the skill.

6 COMPLIANCE CHECKING

In this phase, Skipper derives the skill's behavioral privacy profile (i.e., $\mathcal{B}$) by analyzing the collected logs of conversations, and detects privacy policy noncompliance issues. The main challenge Skipper has to address is to make decision based on the short *utterance-response* conversations.

6.1 Deriving Behavioral Profiles

To infer the types of data collected by the skill (i.e., $\mathcal{B}$.TYPES) is a non-trivial task, while there exist clear indicators to determine the [*Category Vector*] for other three categories (detailed soon in

[2]**Notes on ethics:** we find a residential address located in Los Angles from Google Map, and alter its unit number and street name to an address that does not exist.

Table 6: Keywords examples for log analysis

Data type	Query analysis	Response analysis
Location	"*nearby*", "*near me*", "*nearest*", "*park*", "*position*", "*where can I*", "*where are you*", "*neighborhood*", "*destination*", "*station*", "*region*", "*location*", "*locate*", "*address*", "*city*", "*bus stop*", "*closest*"	"*Brisbane*", "*Australia*" (TC101)
Postcode	"*postcode*", "*post code*", "*zip code*", "*postal code*", "*area code*"	"*4101*", "*04101*" (TC102)
Phone Number	"*phone number*", "*phone*", "*contact number*", "*mobile number*"	"*0450419999*" (TC105)
Birthday	"*brithday*", "*you born*"	"*25th of December*" (TC103)
Name	"*name*", "*full name*", "*sur name*", "*family name*", "*last name*"	"*James Smith*", "*James*", "*Smith*" (TC106)
Email	"*email*"	"*xxx@gmail.com*" (TC104)
Age	"*age*", "*years old*", "*you born*"	"*20 years old*" (TC107)

Section 6.2). Skipper mainly relies on the skill's (active) *queries* and (passive) *responses to the test cases* for this task.

Queries. Queries refer to sentences sent from the skill to the user. As shown in the first and second sentences in our running example (Figure 1), a query can contain the personal information the skill would like to obtain from users.

Responses to test cases. Skipper analyzes the responses of the skill to its privacy-relevant test cases. For example, the skill in our running example (Figure 1) gives the third and fourth sentences to confirm that the user information is recorded by it. Response analysis supplements the query-based detection for the scenarios that the skill does not actively ask for the personal data in their queries but still records such data contained in utterances. This analysis is new in Skipper, compared with existing tester like SkillExplorer [28].

6.1.1 Query Analysis. Skipper extracts queries containing keywords listed in column 2 of Table 6, following the methods of SkillExplorer [28] and SkillDetective [45]. The keywords are adopted from them with mutations based on our test cases. In this way, Skipper gets a candidate set of skills with suspicious data collection behaviors. It then applies grammar-based methods to remove the sentences which are semantically irrelevant to data collection.

Skipper first removes the sentences with the keywords used as proper nouns in a sentence, since they are usually not relevant to data collection. For example, in the sentence "*Welcome back to Refuge Christian Center, Saint Paul, MN. Would you like to hear, Service Times, Location, Phone Number or ask for Help to hear more options*", the keywords "*location*" and "*phone number*" serve as proper nouns as they refer to the particular service information of *Refuge Christian Center*. Skipper uses Spacy [14] to identify the POS of the keywords, and remove the sentence if the keyword is identified as a proper noun.

For the sentences containing noun keywords (e.g., *name*, *email*, *birthday* and *phone number*), we only keep those with the possessive determiner "*your*". This is because such queries indicate the skills' interest in collecting "*your*" (i.e., the user's) information, whereas a typical counter example can be "*Hi, I am the My Favorite Web Designs ChatBot. You can ask for our phone number or address*". Skipper uses Spacy to conduct the constituency parsing on the query and adds the immediate syntactic dependents of the possessive determiner "*your*" into $\mathcal{B}$.TYPES.

6.1.2 Response Analysis. Skipper then processes the responses that follow the utterances containing test cases. We find that it is the common practice for a skill to repeat the collected data in its responses to confirm its data collection. For example, the skill repeats the keyword "*Brisbane*" and birthday as shown in our running example). We therefore keep those skills whose responses contain the keywords listed in the column 3 of Table 6.

We further remove those skills with non-confirmative reply, e.g., "*Sorry, I do not support the currency James-Smith*", "*Your reply, postcode is 4101, is invalid*", and "*The trust at Gmail dot com is not a valid scheme name*". To this end, Skipper determines negative sentences based on a vocabulary list [2]. A response is considered irrelevant to data collection if it is a negative sentence.

After processing both queries and responses, Skipper then generates the $\mathcal{B}$.TYPES for each skill from the remaining sentences through keyword matching (using the same method in Section 4.2.1). For our running example (Figure 1), Skipper adds "*birthday*" and "*location*" into the skill's $\mathcal{B}$.TYPES.

6.2 Violation Detection

With the analysis on the test log, Skipper can determine the non-compliance issues from skills based on the rules defined in Section 2.2. It reports a V1 if $\mathcal{B}.\texttt{TYPES} - \mathcal{D}.\texttt{TYPES} \neq \emptyset$ and a V2 if $\mathcal{B}.\texttt{CHILDREN}.[CTypes] - \mathcal{D}.\texttt{CHILDREN}.[CTypes] \neq \emptyset$ when $\mathcal{B}.\texttt{CHILDREN}.age < \mathcal{D}.\texttt{CHILDREN}.age$. A V3 is reported when Skipper detects from the privacy policy that the skill allows California users to delete collected data ($\mathcal{D}.\texttt{REGIONS}.region$ = California), but the skill neither confirms so upon deletion requests (see Section 5.1.3) nor provides any contact information for users to send deletion requests. Skipper checks V4 by retesting the skill after $\mathcal{D}.\texttt{RETENTION}.period$. It finds the skills that give a welcome message containing "*back*", "*continue*" or "*again*" and then analyzes their sentences using the techniques discussed in Section 6.1 to determine whether they still memerize the collected data.

7 EVALUATION AND LANDSCAPE

We implement Skipper and evaluate it on Amazon Alex skills. Our evaluation aims to study the performance of Skipper. We are also interested in understanding the landscape of privacy policy compliance in real-world VPA apps. Therefore, we target to answer the following three research questions (RQs).

RQ1. What is Skipper's performance in identifying privacy policy noncompliance issues in terms of the four privacy policy categories?
RQ2. Based on Skipper's findings, what is the *status quo* of privacy policy compliance in existing skills? Do they follow their declared privacy profiles at runtime?
RQ3. From the identified noncompliance cases, what characteristics can be summarized?

7.1 RQ1: Performance Evaluation

Before conducting a large-scale auditing on all skills available online, we first study the performance of Skipper. Since there is not a benchmark available in the literature, we proceed with constructing one for our study. For the TYPES category, we randomly select 200 skills from 20 categories. Without losing representativeness, the number of selected skills from each category are proportional to the skill distributions among categories. Out of these 200 skills, 31 provide valid privacy policy documents (although 64 provide links). Since only a small proportion of skills have CHILDREN and REGIONS policies (See Table 3), we randomly select 50 skills to evaluate each

Table 7: SKIPPER's performance on the benchmark dataset

	V1† TYPES					V2 CHILDREN		V3 REGIONS		V4 RETENTION	
	Providing Privacy Policy			No Privacy Policy							
	POS	NEG		POS	NEG	POS	NEG	POS	NEG	POS	NEG
	$\mathcal{B}-\mathcal{D}\neq\emptyset$	$\mathcal{B}\subset\mathcal{D}$	$\mathcal{B}=\mathcal{D}$	$\mathcal{B}\neq\emptyset$	$\mathcal{B}=\emptyset$						
GroundTruth	1	30	0	9	160	4	46	4	46	3	47
SKIPPER	1	28	0	13	158	4	46	4	46	3	47
FP	4					0		0		0	
TP	10					4		4		3	
FN	0					0		0		0	
TN	186					46		46		47	
FP rate	2.11%					0%		0%		0%	
Precision	71.42%					100%		100%		100%	
Recall	100%					100%		100%		100%	

† In this table, we abuse $\mathcal{B}$ to stand for $\mathcal{B}$.TYPES, and $\mathcal{D}$ for $\mathcal{D}$.TYPES. POS: Positive, NEG: Negative. TP: true positive, FP: False positive, TN: true negative, and FN: false negative.

Table 8: Break-down results of V1

Category	No Privacy Policy			Providing Privacy Policy without TYPES			Providing Privacy Policy with TYPES			
	$\mathcal{B}\neq\emptyset$	$\mathcal{B}=\emptyset$	Total	$\mathcal{B}\neq\emptyset$	$\mathcal{B}=\emptyset$	Total	$\mathcal{B}\subset\mathcal{D}$	$\mathcal{B}=\mathcal{D}$	$\mathcal{B}-\mathcal{D}\neq\emptyset$	Total
Car (172)	1 (0.79%)	125	126	1 (9.09%)	10	11	33	0	2 (5.71%)	35
Business (3824)	61 (1.92%)	3124	3185	4 (2.82%)	138	142	486	0	11 (2.21%)	497
Education (6344)	126 (2.11%)	5853	5979	3 (3.03%)	96	99	254	1	12 (4.51%)	266
Food (1337)	38 (3.13%)	1178	1216	0 (0.00%)	17	17	104	0	0 (0.00%)	104
Game (6408)	95 (1.56%)	6008	6103	3 (4.05%)	71	74	226	0	5 (2.16%)	231
Kids (3760)	5 (0.14%)	3659	3664	0 (0.00%)	23	23	73	0	0 (0.00%)	73
Life (6211)	40 (0.71%)	5633	5673	4 (3.88%)	99	103	434	0	1 (0.23%)	435
Local (1194)	62 (5.69%)	1027	1089	3 (11.11%)	24	27	74	0	4 (5.13%)	78
Movie (862)	4 (0.48%)	829	833	1 (12.50%)	7	8	21	0	0 (0.00%)	21
Music (6386)	58 (0.96%)	5984	6042	3 (3.66%)	79	82	258	0	4 (1.53%)	262
News (6396)	26 (0.44%)	5844	5870	0 (0.00%)	149	149	376	0	1 (0.27%)	377
Novelty (3472)	29 (0.85%)	3384	3413	1 (5.00%)	19	20	38	0	1 (2.56%)	39
Product (5206)	99 (2.05%)	4736	4835	4 (4.76%)	80	84	274	0	13 (4.53%)	287
Shopping (408)	5 (1.79%)	274	279	0 (0.00%)	21	21	107	0	1 (0.93%)	108
Home (2818)	12 (0.81%)	1461	1473	1 (0.42%)	235	236	1102	1	7 (0.63%)	1109
Social (1882)	25 (1.39%)	1777	1802	0 (0.00%)	11	11	65	0	4 (5.80%)	69
Sports (1733)	29 (1.77%)	1612	1641	0 (0.00%)	23	23	69	0	0 (0.00%)	69
Travel (1197)	119 (10.76%)	987	1106	4 (18.18%)	18	22	69	1	0 (0.00%)	69
Utilities (1158)	34 (3.10%)	1064	1098	1 (6.67%)	14	15	39	1	6 (13.33%)	45
Weather (737)	21 (3.07%)	663	684	2 (12.50%)	14	16	37	0	0 (0.00%)	37
Total (61505)	889 (1.58%)	55222	56111	35 (2.96%)	1148	1183	4139	4	72 (1.71%)	4211

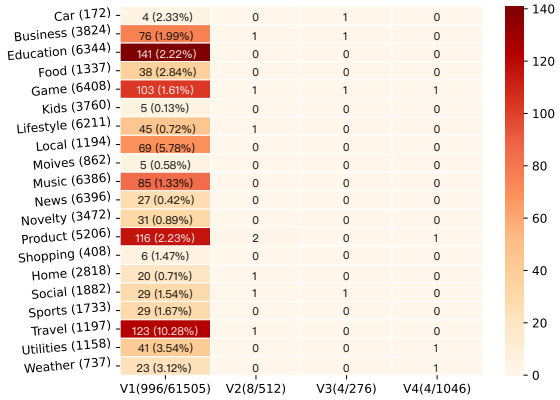

Figure 4: Distribution of skills w/ ≥1 noncompliance issues

category from the skills that have relevant policies. We evaluate all skills that have data collection behaviors for their RETENTION policies, as this category relies on the data collection.

We apply SKIPPER to check the skills in our benchmark. After SKIPPER reports its findings, we resort to manual efforts to create the ground truth. To avoid bias, we engage two volunteers from our research lab to annotate the benchmark skills. They have never been introduced to the internals of SKIPPER. For each benchmark skill, they are asked to read its privacy policy document and test logs, and label the noncompliance issues of the four categories. After independent annotation, they discuss with each other to resolve disagreement. Then, we check SKIPPER's reports against their annotations. The results are listed in Table 7.

In general, SKIPPER achieves a high rate of detecting violations notably with recalls of 100% across all categories, meaning that SKIPPER is able to detect nearly all violations despite of false positives of V1 that need further confirmation (discussed soon). For V1, SKIPPER falsely identifies four skills as positive, while for the rest of violation (V2-V5), SKIPPER achieves exactly the same results as the ground truth. We break down the scenarios in Column 2-6 of Table 7 to investigate the false positive cases in V1. The four false positives are caused by the inaccuracy in query analysis. For example, in a skill's query "*Welcome to the Priced Rite Auto skill. Ask what are the latest vehicles? What is the phone number? What is your location? or what are your hours?*", the skill is expecting the user to repeat "*What is your location?*" for the service enquiry rather than actively asking for the user's location. This challenges the existing NLP techniques as it is hard for a machine learning algorithm in such cases to infer the correct intention. We therefore recommend that further confirmation should be conducted on the detected positive cases when SKIPPER is used in practice.

7.2 RQ2: Privacy Policy Compliance Landscape

After benchmarking, we conduct a large-scale study with SKIPPER to understand the *status quo* of privacy policy compliance by the skills in the wild. SKIPPER enumerates all 61,505 in the UQ-AAS21 dataset to test. Overall, 24,250 skills are runnable. From them, SKIPPER detects 1,012 noncompliance cases (4.2% of the runnable skills) in total, with 996 in V1, 8 (out of 512 testable skills that have CHILDREN policies) in V2, 4 (out of 276 testable skills with REGIONS) in V3, and 4 (out of 1046 with data collection behaviors) in V4. Figure 4 shows the occurrence of V1-V4 across the 20 categories on Alexa store.

TYPES (V1). This issue dominates all skill categories, suggesting the pervasiveness of under-disclosed personal data collection in skills. Table 8 breaks down the findings according to the skill's declared and behavioral profiles. It can be found that most issues happen when skills request for users' personal information without providing privacy policies (889/996). Other 35 skills provide privacy policies but miss the TYPES policies, and 72 skills fail to declare the correct types of personal information in their privacy policies.

CHILDREN (V2). SKIPPER detects 8 noncompliance issues in V2 among 512 testable skills (out of 1,643 that provide CHILDREN policies in their document), including two in the Product category, and one in each of Business, Travel, Home, Game and Life. Such violations shall raise awareness to skill developers as there is specific restriction on children's data protection in regulations.

REGIONS (V3). SKIPPER reports 4 noncompliance issues in V3 out of 276 testable skills. They all fail to confirm SKIPPER's requests of personal data deletion while they declare to satisfy CCPA and provide users with the "*Right to delete*". We additionally check their privacy policy documents. All provide contact information, but three provide it outside the REGIONS section, and the other provides an invalid link.

We further investigate this issue from the perspective of privacy policy documents, given that the skill may ask the user to contact the developer for deleting data. We extract developer contact information (including their contact numbers, email addresses and post addresses) from those skills which include REGIONS section

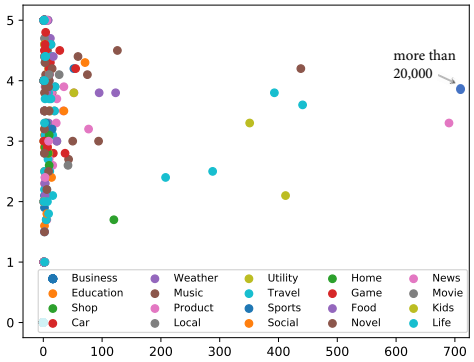

Figure 5: Rating scores (Y axis) and the number of ratings (X axis) of noncompliance skills

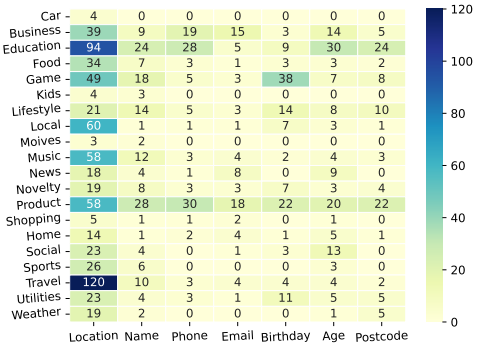

Figure 6: Noncompliance issue distribution among data types

(1,299 in total). We find that 906 (70%) provide developer's contact information inside the REGIONS paragraph. For the remaining documents, 171 (13%) provides the information outside the REGIONS paragraphs, and 222 (17%) do not provide any contact information throughout the entire document.

RETENTION (V4). Skipper detects 4 in V4 among 1046 skills which collect users' personal data. Take one of them as an example. The skill, which has collected the user's birthday, says "*Welcome back. It looks like there are 339 days until your 13th birthday*" as an opening sentence when it is accessed the second time. However, it fails to include the RETENTION policies in its privacy policy document to declared its behavior of data retention.

We also examine the status of skills' RETENTION policies. We find that only around 300 skills (among 5,829 that provide privacy policy documents) clearly state their retention period. Another 796 skills include RETENTION policies but do not provide a specific time period, and other 4,829 skills do not include a RETENTION section.

7.3 RQ3: Characterize noncompliance skills

We conduct a study on the detected noncompliance cases to reveal their characteristics. We first check whether their users are aware of the noncompliance issue. Figure 5 shows the rating scores and the number of ratings of all noncompliance skills. Most of them have a relatively high score. We search through the user comments of these skills and find the users seldom mention the privacy policy compliance. This suggests that the public have not become vigilant in checking the privacy policy compliance.

We further analyze the statistics according to the types of personal data violations in Figure 6. Skipper detects the most violation cases in collecting users' location (689), followed by name (158) and postcode (133). Violations in terms of phone number collection are the least (73). For violation cases in collecting location data type, travel category has the highest number (120), the next one is education category (94). Skills in product category have the highest violation cases in collection name data type.

We also check the correlation between skills' privacy policies and their requested permissions. We find that half (2,666) of all 5,829 skills that have a valid privacy policy have mismatching issues between their policies and requested permissions. Among them, 410 request permissions but the requested data types do not match the declared types in their privacy policies. The other 2,256 declare to collect user data (i.e., $\mathcal{D}$.TYPES $\neq \emptyset$), but request no permission.

8 DISCUSSIONS

8.1 Implications

Our findings should highlight the urgency to identify and avoid the privacy policy noncompliance issues.

8.1.1 Store Operators. The main reason for the massive number of noncompliance issues is that the skill store does not enforce a strict requirement on releasing and complying with the privacy policy documents. The small proportion (7,513/65,195) of skills that provide accessible privacy policies is worrisome, and the quality of the available privacy policy documents varies. Most noncompliance issues (889/996) happen when skills request users' personal data without providing any privacy policy document. We recommend the store operators should mandate a privacy policy when a skill is released through the store. In addition, the operators should consider providing guidelines and appropriate templates to facilitate the development of privacy policies by developers.

The operators should become aware of the high feasibility of runtime information gathering due to the conversational nature of VPA apps. The existing permission system [21] may be sufficient in the mobile app ecosystem because the APIs are the main (if not the only) channel for apps to acquire user data, but it has obvious limitations in preventing such issues in the VPA context. Dishonest skills could easily bypass the permission checking and collect user data through conversations. Therefore, we recommend the operators should mandate skills to disclose their data handling practice in a more observable way. The operators should also include the compliance checking in their vetting process.

8.1.2 Users. Awareness should also be raised among the end users. The users should carefully check the privacy policy document of the skill to understand its data handling practice before starting using it. Besides, the users should also look out for runtime information gathering behaviors of the skill. They should pay attention when the skill queries information, but this behavior has not been disclosed in the privacy policy document or the collected information is irrelevant to the functionality of the skill.

We especially remind parents of the protection of children's data. A recent study [30] shows that kids can access and spend a long time on VPA apps. Although Amazon has enforced strict policies on children users, Skipper still finds a few violations on CHILDREN category. Thus, we suggest parents pay special attention to the skills used by their kids.

8.2 Limitations

Skipper focuses on the privacy policy noncompliance issues arising when VPA apps collect, use and store user data. To the best of our knowledge, this is the first work that conducts a systematic large-scale analysis. However, as the first attempt in this area, current work of Skipper carries several limitations that could be addressed in future work.

First, Skipper's privacy policy document interpretation relies on NLP and machine learning techniques to derive the declared privacy profile. As shown in Section 4, there is still a failure rate of 15%. We call for domain-specific NLP techniques to facilitate the interpretation of legal documents like privacy policies and terms of use. Second, due to the unavailability of skill internals, Skipper is unable to scan its code base for data handling operations and has to determine the behavioral privacy profile based on the skill's conversation with the user. This can be inaccurate. A malicious skill may intentionally escape Skipper's detection by giving no affirmative response (see Section 6.2), for example, saying "*I don't understand it*" but actually storing the sensitive data given by the user, or storing collected data permanently but never appearing to have store the data. In contrast, an honest skill may query information for benign purpose (see the assumption in section 2.2) and still be flagged as non-compliant. This unavailability may cause the third limitation of Skipper that many policy categories remain untestable, for example, whether the skill has shared collected data with a third party. The malicious skill can also invoke APIs to access user data. Fourth, Skipper does not focus on the enhancement of the test engine. One with higher coverage rate would trigger more privacy-related skill behaviors. Fifth, in the response analysis, Skipper can derive behavioral profiles only when the skill gives positive responses or repeats keywords. However, if the skill intentionally escapes this, Skipper could miss them out.

9 RELATED WORK

Skipper is related to the analysis of privacy policy quality and privacy testing. In this section, we summarize existing studies related to them.

Analysis of Skill Data Handling Practices. Lentzsch et al. [31] conduct a large-scale study of Alexa skill ecosystem, including the skill vetting process, squatting attacks, permissions, etc. They investigate the availability of privacy policy links and the compliance between the privacy policy and the permissions, and reveal that only 24.2% skills provide privacy policies links. Edu et al. [25, 26] also study the skill's data disclosure practices and compliance with the permissions, based on a large-scale skill dataset across three years (2019-2021). In terms of the quality analysis, Skipper confirms their findings and the revealed facts of unsatisfactory data handling practices in common. Besides that, Skipper further examines detailed policy compliance including the data types, child data protection, region-specific policies and data retention. Its fine-grained analysis reveals insights and provide users and developers detailed information regarding the privacy policy compliance.

Extracting Claims from Privacy Policy Documents. There is a line of research [25, 34, 41] proposed to extract applications' claims on their data handling practice from the privacy policy documents. Edu et al. [25] develop *Skillvet*, which uses binary models to process sentences and derives corresponding permissions. It covers the data types listed in Alexa permission system and demonstrates high classification accuracy. Liao et al. [34] also conduct sentence-level processing. They first identify basic phrases and construct keyword dictionaries, and then process documents by locating sentences related to data processing to derive corresponding data types. This method may have limitations when dealing with context-sensitive situations. For example, the sentence "*if you have any questions, please send an email to us*" in a *contact us* section matches the phrase format but it does not mean the skill collects email data. Torre et al. [41] transfer privacy policy documents into conceptual models by keyword-based classification, and use the models to check the compliance with GDPR. It may not be directly applicable as Skipper needs the exact data types.

Table 9: A comparison among Skipper and two studies

	Target privacy policies†				Policy-driven test cases	Detection		#noncompliance issues found
	T	C	REG	RET		Q	R	
SkillExplorer	●	○	○	○	○	●	○	N/A‡
SkillDetective	●	◐§	○	○	○	●	○	623
Skipper	●	●	●	●	●	●	●	1,012

† T: TYPES, C: CHILDREN, REG: REGIONS and RET: RETENTION. Q: Detection based on queries, R: Detection based on responses to policy-driven test cases. ●/○: covered/not covered.
‡ SkillExplorer does not specifically target privacy policy compliance checking.
§ SkillDetective checks the children data handling of the skills in *kids* category. It does not target other skills with CHILDREN policies.

Skill Privacy Testing. Guo et al. [28] propose the first systematic study on Alexa skills' behaviors through testing. They develop SkillExplorer which uses a grammar-based i-tree to recognize query types and uses a chatbot to drive the black-box testing. A couple of studies [26, 30, 33, 45] inspired by SkillExplorer use black-box testing to explore skills' data collection behaviors. SkillDetective [45] analyzes the data collection behaviors against privacy policies. SkillBot [30] specifically targets children-risky contents in skills. Table 9 presents a brief comparison between Skipper with SkillDetective and SkillExplorer. In general, Skipper targets those testable policies, and constructs policy-related test cases and analyzes skills' responses to them, so it can detect more noncompliance issues. In terms of children's data protection, Skipper can check the four types of noncompliance issues in all skills with CHILDREN policy, while SkillDetective and SkillBot specifically target the skills of the *kids* category (see the row of *kids* in Table 8).

10 CONCLUSION

Skipper aims to automatically identify privacy policy noncompliance issues from the VPA apps (i.e., *skills*) of Amazon Alexa, the most popular VPA service. The main idea of Skipper is to derive the skill's declared privacy profile (i.e., $\mathcal{D}$) from the data protection practices it describes in the privacy policy document and its behavioral privacy profile (i.e., $\mathcal{B}$) based on black-box testing. Our work reveals the *status quo* of the privacy policy compliance in these emerging voice-based VPA apps. We remark that Skipper is a preliminary work in the direction of privacy policy compliance auditing, and more future studies are desirable to cope with the challenges we report.

Availability. The source code of Skipper and relevant artifacts including executable examples and benchmark logs are available on GitHub [12] and Zenodo [13].

ACKNOWLEDGMENTS

We thank our anonymous shepherd and reviewers for their insightful comments to improve this manuscript. The authors from University of Queensland are supported in part by UQ NSRSG grant and Australian Research Council under CE200100025. The authors from Nanjing University are supported in part by the Leading-edge Technology Program of Jiangsu Natural Science Foundation (No. BK20202001), the National Natural Science Foundation of China (No. 62172200 and No.62032010), and the Fundamental Research Funds for the Central Universities (No. 020214380094). IIE authors are supported in part by NSFC U1836211, Beijing Natural Science Foundation (No. M22004), the Anhui Department of Science and Technology under Grant 202103a05020009, Youth Innovation Promotion Association CAS, Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] 1998. *Children's Online Privacy Protection Rule ("COPPA").* https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa
[2] 2012. *Negative Vocabulary Word List.* https://www.enchantedlearning.com/wordlist/negativewords.shtml
[3] 2018. *GDPR Article 13 - Information to be provided where personal data are collected from the data subject.* https://gdpr-info.eu/art-13-gdpr/
[4] 2020. *Create Your Amazon Household.* Retrieved January 7, 2022 from https://www.amazon.com/gp/help/customer/display.html?nodeId=GYLAACCNR8G3VVRM
[5] 2020. *General Data Protection Regulation(GDPR).* Retrieved January 26, 2022 from https://gdpr-info.eu/
[6] 2020. *Google Fined $57M by Data Protection Watchdog Over GDPR Violations.* https://digitalguardian.com/blog/google-fined-57m-data-protection-watchdog-over-gdpr-violations
[7] 2020. *NLTK Documentations.* Retrieved January 26, 2022 from https://www.nltk.org
[8] 2021. *Amazon Gets Record $888 Million EU Fine Over Data Violations.* https://www.bloomberg.com/news/articles/2021-07-30/amazon-given-record-888-million-eu-fine-for-data-privacy-breach
[9] 2021. *Colorado Privacy Act ("CPA").* https://www.consumerprivacyact.com/colorado-privacy-act-cpa/
[10] 2022. *Amazon Developer Console.* Retrieved April 1, 2022 from https://developer.amazon.com/alexa/console/ask
[11] 2022. *California Consumer Privacy Act.* https://oag.ca.gov/privacy/ccpa
[12] 2022. *SKIPPER (Source Code).* https://github.com/UQ-Trust-Lab/SKIPPER
[13] 2022. *SKIPPER (Zenodo).* https://doi.org/10.5281/zenodo.7045277
[14] 2022. *SpaCy Documentations.* Retrieved April 1, 2022 from https://spacy.io
[15] 2022. *To detect the language of the text.* https://pypi.org/project/langdetect/
[16] 2022. *Understand Smart Home Skills.* Retrieved January 7, 2022 from https://developer.amazon.com/en-US/docs/alexa/smarthome/understand-the-smart-home-skill-api.html
[17] 2022. *Virtual Assistant Technology - Statistics & Facts.* Retrieved April 1, 2022 from https://www.statista.com/topics/5572/virtual-assistants/#topicHeader__wrapper
[18] 2022. *WordNet Documentations.* Retrieved January 26, 2022 from https://wordnet.princeton.edu/
[19] 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 *Proceedings of the 28th USENIX Conference on Security Symposium (Usenix Security)*. 585–602.
[20] California Consumer Privacy Act (CCPA). 2022. *E. REQUESTS TO DELETE PERSONAL INFORMATION.*
[21] Amazon Developer Documentation. 2022. *Configure Permissions for Customer Information in Your Skill.* https://developer.amazon.com/en-US/docs/alexa/custom-skills/configure-permissions-for-customer-information-in-your-skill.html
[22] Amazon Developer Documentation. 2022. *Policy Testing.* https://developer.amazon.com/en-US/docs/alexa/custom-skills/policy-testing-for-an-alexa-skill.html
[23] Amazon Developer Documentation. 2022. *Security Testing for an Alexa Skill.* https://developer.amazon.com/en-US/docs/alexa/custom-skills/security-testing-for-an-alexa-skill.html
[24] Amazon Developer Documentation. 2022. *Test with the Alexa Simulator.* https://developer.amazon.com/en-US/docs/alexa/devconsole/alexa-simulator.html
[25] Jide Edu, Xavi Ferrer Aran, Jose Such, and Guillermo Suarez-Tangil. 2021. SkillVet: Automated Traceability Analysis of Amazon Alexa Skills. *IEEE Transactions on Dependable and Secure Computing* (2021).
[26] Jide Edu, Xavier Ferrer Aran, Jose Such, and Guillermo Suarez-Tangil. 2022. Measuring Alexa Skill Privacy Practices across Three Years. In *The Web Conference (WWW)*. ACM.
[27] Google. 2022. *Manage your Google app permissions.* https://support.google.com/websearch/answer/10000369?hl=en&ref_topic=6032684
[28] Zhixiu Guo, Zijin Lin, Pan Li, and Kai Chen. 2020. Skillexplorer: Understanding the behavior of skills in large scale. In *29th USENIX Security Symposium (USENIX Security)*. 2649–2666.
[29] Deepak Kumar, Riccardo Paccagnella, Paul Murley, Eric Hennenfent, Joshua Mason, Adam Bates, and Michael Bailey. 2018. Skill squatting attacks on Amazon Alexa. In *27th USENIX security symposium (USENIX Security)*. 33–47.
[30] Tu Le, Danny Yuxing Huang, Noah Apthorpe, and Yuan Tian. 2022. SkillBot: Identifying Risky Content for Children in Alexa Skills. *ACM Trans. Internet Technol.* 22, 3, Article 79 (jul 2022), 31 pages. https://doi.org/10.1145/3539609
[31] Christopher Lentzsch, Sheel Jayesh Shah, Benjamin Andow, Martin Degeling, Anupam Das, and William Enck. 2021. Hey Alexa, is this skill safe?: Taking a closer look at the Alexa skill ecosystem. In *28th Annual Network and Distributed System Security Symposium (NDSS)*.
[32] Linyi Li, Zhenwen Li, Weijie Zhang, Jun Zhou, Pengcheng Wang, Jing Wu, Guanghua He, Xia Zeng, Yuetang Deng, and Tao Xie. 2020. Clustering Test Steps in Natural Language toward Automating Test Automation. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (FSE)*. 1285–1295.
[33] Suwan Li, Lei Bu, Guangdong Bai, Zhixiu Guo, Kai Chen, and Hanlin Wei. 2022. Guided Model-based VUI Testing of VPA Apps. In *37th IEEE/ACM International Conference on Automated Software Engineering (ASE'22)*.
[34] Song Liao, Christin Wilson, Long Cheng, Hongxin Hu, and Huixing Deng. 2020. Measuring the Effectiveness of Privacy Policies for Voice Assistant Applications. In *Annual Computer Security Applications Conference* (Austin, USA) *(ACSAC '20)*. Association for Computing Machinery, New York, NY, USA, 856–869. https://doi.org/10.1145/3427228.3427250
[35] Shuang Liu, Renjie Guo, Baiyang Zhao, Tao Chen, and Meishan Zhang. 2020. APPCorp: A Corpus for Android Privacy Policy Document Structure Analysis. *CoRR* abs/2005.06945 (2020). arXiv:2005.06945
[36] Shuang Liu, Baiyang Zhao, Renjie Guo, Guozhu Meng, Fan Zhang, and Meishan Zhang. 2021. Have You Been Properly Notified? Automatic Compliance Analysis of Privacy Policy Text with GDPR Article 13. In *Proceedings of the Web Conference (WWW)*. Association for Computing Machinery, New York, NY, USA, 2154–2164. https://doi.org/10.1145/3442381.3450022
[37] Rozita Dara Niharika Guntamukkala and Gary Grewal. 2015. A machinelearning based approach for measuring the completeness of online privacy policies. In *In 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE.* 289–294.
[38] Jason DM Rennie. 2001. Improving multi-class text classification with naive Bayes. (2001).
[39] Roman Schlegel, Kehuan Zhang, Xiao-yong Zhou, Mehool Intwala, Apu Kapadia, and XiaoFeng Wang. 2011. Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. 17–33.
[40] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D Breaux, and Jianwei Niu. 2016. Toward a framework for detecting privacy policy violations in android application code. In *Proceedings of the 38th International Conference on Software Engineering (ICSE)*. 25–36.
[41] Damiano Torre, Sallam Abualhaija, Mehrdad Sabetzadeh, Lionel Briand, Katrien Baetens, Peter Goes, and Sylvie Forastier. 2020. An AI-assisted Approach for Checking the Completeness of Privacy Policies Against GDPR. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*. 136–146. https://doi.org/10.1109/RE48521.2020.00025
[42] Payton Walker and Nitesh Saxena. 2021. Evaluating the Effectiveness of Protection Jamming Devices in Mitigating Smart Speaker Eavesdropping Attacks Using Gaussian White Noise. In *Annual Computer Security Applications Conference (ACSAC)*. 414–424.
[43] Xiaoyin Wang, Xue Qin, Mitra Bokaei Hosseini, Rocky Slavin, Travis D Breaux, and Jianwei Niu. 2018. Guileak: Tracing privacy policy claims on user input data for android applications. In *Proceedings of the 40th International Conference on Software Engineering (ICSE)*. 37–47.
[44] Fuman Xie, Yanjun Zhang, Hanlin Wei, and Guangdong Bai. 2022. UQ-AAS21: A Comprehensive Dataset of Amazon Alexa Skills. In *17th International Conference Advanced Data Mining and Applications (ADMA)*. 159–173.
[45] Jeffrey Young, Song Liao, Long Cheng, Hongxin Hu, and Huixing Deng. 2022. SkillDetective: Automated Policy-Violation Detection of Voice Assistant Applications in the Wild. In *31st USENIX Security Symposium (USENIX Security)*.
[46] 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)*. IEEE, 538–549.
[47] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. 2019. Dangerous skills: Understanding and mitigating security risks of

voice-controlled third-party functions on virtual personal assistant systems. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1381–1396.

[48] Nan Zhang, Kan Yuan, Muhammad Naveed, Xiaoyong Zhou, and Xiaofeng Wang. 2015. Leave Me Alone: App-Level Protection against Runtime Information Gathering on Android. In *2015 IEEE Symposium on Security and Privacy (SP)*. 915–930.

[49] Yangyong Zhang, Lei Xu, Abner Mendoza, Guangliang Yang, Phakpoom Chinprutthiwong, and Guofei Gu. 2019. Life after speech recognition: Fuzzing semantic misinterpretation for voice assistant applications. In *Proc. of the Network and Distributed System Security Symposium (NDSS)*.

[50] 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 *Proc. of the Network and Distributed System Security Symposium (NDSS)*.