Assessing and Measuring the Privacy Practices of Voice Assistant Applications

Edu, Jide

Awarding institution:
King's College London

Assessing and Measuring the Privacy Practices of Voice Assistant Applications

Jide Edu

Supervisor: Prof. Jose Such
Dr. Guillermo Suarez-Tangil

Department of Informatics
King's College London

This dissertation is submitted for the degree of
Doctor of Philosophy

September 2022

I would like to dedicate this thesis to my lovely parents

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements.

Jide Edu
September 2022

Acknowledgements

I sincerely want to thank my supervisors, Prof. Jose Miguel Such and Dr. Guillermo Suarez-Tangil, for their help, guidance, support, and motivation towards the success of my PhD program. I appreciate your patience, excellent suggestion, encouragement, mentoring, and time without which it would be impossible to complete this work. I am privileged to have had the opportunity of working under your stewardship, and I am eternally grateful. I look forward to continuing working and learning from you.

I want to say a big thank you to Dr Xavier Ferrer-Aran, for your intellectual discussions and support. It is great having you around, and I am forever grateful.

I would like to offer my special thanks to my examiners, Dr Hamed Haddadi and Prof Julio Hernandez-Castro, I appreciate your time, effort and insightful comments that have helped me improve this thesis.

I would also like to express my gratitude to my parents, my wife and my children. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

At King's, I met many other colleagues, BSc, MSc, PhD students, and Professors, all worthy of being acknowledged for dedicating their time to ensure that I completed this work. To King's and all the informatics department administrative staff, I thank you for your support and services. I am grateful to have been part of such a supportive community.

I am deeply grateful to the Government of the Federal Republic of Nigeria through the Petroleum Technology Development Fund Agency (PTDF) for sponsoring and providing financial support for this work. I remain indebted to them.

Finally, I would like to thank God almighty for his mercy, guidance, and blessing upon me, my friends, family, and loved ones.

Abstract

Smart Personal Voice Assistants (SPA) are fast becoming popular with the widespread introduction of desktop, phone and home assistants. Over a hundred million users now utilise SPA like Alexa, Siri, Google Assistant, Bixby and Cortana every day, and SPA devices have been sold in massive numbers. However, recent security and privacy incidents involving SPA like Alexa recording a private conversation and sending it to a random contact have increased users' concerns about the security and privacy of these assistants. This thesis studies the security and privacy issues of SPA. In particular, the risks associated with the skills (voice applications) they leverage to extend and expand their functionality. Firstly, we present a classification of SPA security and privacy issues and use it to systematically map current attacks and countermeasures to different architectural elements. We show that those elements expose SPA to various risks, such as the complexity of their architecture, the AI features, the wide range of underlying technologies, and the open nature of the voice channel they use.

We then conduct a systematic study of SPA third-party skills as this is one of the architectural elements offering a large attack surface. In particular, we study the permission model SPA providers offer to developers and investigate how third-party skills use them to collect personal data. We further design a methodology that systematically identifies potential privacy issues in the third-party skills by analysing the traceability between the permissions and the data practices stated by developers. In addition, we propose a highly accurate system to automate the traceability analysis at scale. Furthermore, we perform a longitudinal measurement study of the Amazon Alexa skills across the marketplaces for three years to demystify developers' data practices and present an overview of the third-party skill ecosystem. Finally, we present an open tool that allows proactive audit of data collection practices in emerging technologies like SPA. The overall study resulted in two new datasets for smart assistants privacy assessment evaluation: the traceability-by-policy dataset (TBPD) and the permission-by-sentence dataset (PBSD). All these aim to contribute to the collective effort towards establishing secure, privacy-aware assistants.

Table of contents

List of abbreviations

The next list describes several abbreviations that will be later used within the body of the document

API	Application Programming Interface
$APP-350$	350 Android Privacy Policies
ASR	Automatic Speech Recognition
CSS	Cascading Style Sheet
$HTML$	Hypertext Markup Language
ML	Machine Learning
NLP	Natural Language Processing
$OBJ1-OBJ4$	Research Objective 1-4
$PBSD$	Permission-By-Sentence Dataset
RQ	Research Question
SPA	Smart Personal Voice Assistants
$TBPD$	Traceability-By-Policy Dataset
$VAPA$	Voice Assistant Privacy Assessment Tool
$\mathcal{D}$	Levenshtein Distance
t	Traceability Result

List of figures

List of tables

Chapter 1

Introduction

Human-computer interaction (HCI) has traditionally been conducted in the form of different types of peripheral devices such as the keyboard, mouse and, most recently, tactile screens. This has been so because computing devices could not decode the meaning of our words, let alone understand our intent. However, the paradigm has shifted over the last few years, as we witnessed the rapid development of voice technology in many computing applications. Since voice is one of the most effective and expressive communication tools, voice technology is changing the way in which users interact with devices and the manner they consume services.

One of the most significant innovations that use voice technology is Smart Personal Voice Assistants (SPA). SPA are changing how users interact with technology and the way they consume services [22, 66, 203], driving new experiences and expectations [1]. Advances in natural language processing enable SPA to take voice commands, which they process using machine learning to deduce users' intentions and fulfil users' requests. They offer hands-free and eye-free operations, allowing users to perform diverse activities using voice commands while concentrating on other tasks. Besides offering users the benefit of a quick interaction — humans speak faster than they type [197], using voice for HCI is considered more natural [109] when compared to other interfaces like keyboard and mouse. Not to mention the more substantial social presence offered to users when they hear synthesised speeches very much like their own as responses from this technology [132].

SPA are rapidly becoming standard features in homes integrated into smart speakers like Amazon Echo, Google Home, Apple HomePod [190], smart TVs and many other smart devices [189]. According to a recent report, nearly 90 million adults in the US use at least one SPA device [122], and the adoption is expected to rocket even further as SPA integrate into smartphones [121] or as a chatbot [120]. There are several features that contribute to the popularity of SPA. SPA are quite different from early voice-activated technologies that only work with small inbuilt commands and responses. Instead, SPA use Internet services and benefits from recent advances in Natural Language Processing (NLP), which allow them to handle a wide range of commands and questions. They enable a playful interaction, making their use more engaging [139]. They are assigned a name and a gender, which encourages users to personify them and therefore interact with them in a human-like manner [164].

SPA incorporate voice-driven applications generally developed by third-parties, referred to as *skills* in Amazon Alexa and *actions* in Google Assistant.[1] Skills are a fundamental component of the SPA system. Like in mobile apps, skills play an essential role in defining the SPA capabilities by offering a wide range of services. The entire skills ecosystem provides an environment that allows the user to run more complex functions such as calendar management, shopping, and music playback. However, and in contrast to mobile apps, skills don't run on any user-controlled device, as discussed later. Skills have multiplied in recent years, and the number keeps growing daily. For instance, the Amazon Alexa skill ecosystem has grown from just 135 skills in early 2016 [180] to over 100k skills by late 2020, as details later in Chapter 3. This rapid surge in numbers can be attributed to the continuous proliferation of SPA worldwide.

SPA leverage skills to expand and extend their functionality, allowing them to maintain shopping and to-dos lists, purchase goods and food, play audiobooks, play games, stream music, radio and news, set timers, alarms and reminders [210], get recipe ideas, send messages [113] and many more depending on their usage context [240, 96]. They are also used to manage other IoT and smart home devices [1, 231] bringing the smart home into one verbally controlled system [147]. With the continuous proliferation and the rapid growth of SPA, we are now approaching an era when SPA will not only be manoeuvring our devices at home but also replacing them in many cases. For instance, many SPA are now able to make phone calls, which positions them as a communicating device, and a likely alternative to landlines phones in the future, and some SPA are also equipped with display interface for watching videos/movies and smart home cameras directly in the SPA devices [11].

[1] Note that, for ease of exposition, we adopt Amazon's terminology of voice applications — *Skills*, but these may be called differently in other SPA platforms.

1.1 Problem Description

As SPA become increasingly popular [170], the most sought-after features expose them to various risks. Some of those features are the open nature of the voice channel they use, the complexity of their architecture, the AI features they rely on, and the use of a wide range of different technologies. It is paramount to understand the underlying risks behind their use and fathom how to mitigate them. While most of these assistants have incorporated some security and privacy mechanisms in their design, there is still a significant number of security and privacy challenges that need to be addressed. This is all the more important because SPA carry out distinct roles and perform various functions in single and multi-user environments, particularly in an intimate domain like homes. Since users co-locate with this technology, it also has an impact on the changes in their neighbouring environment [186]. In fact, there have already been reported security and privacy incidents in the media involving SPA, such as the case of an Amazon Alexa recording an intimate conversation and sending it to an arbitrary contact [240]. A recent study also showed how SPA could be misactivated by conversations that are not intended for it [66].

Furthermore, the third-party skills integrated by SPA also widen their attack surface. Unlike mobile apps, skills do not run on any user-controlled device. Instead, skills either run in the Amazon cloud (e.g., as an AWS Lambda function) or in a server controlled by the developer [19]. This could allow malicious developers to sneak malicious code into their software via the application backend [212]. Developers could likewise manipulate skills to covertly introduce misperception about reported events [206] or impersonate another skill [252, 127]. Moreover, skills allow users to interact with their services and require the exchange of personal data. The enormous amount of personal data skills could collect also opened up the SPA to another avenue of attack just like other IoT devices [37].

Users are concerned about the security and privacy of these assistants [77, 68]. In the absence of better technical security and privacy controls, users are implementing workarounds like turning off the SPA when they are not using it [1, 130]. Unfortunately, several mitigating techniques proposed in various studies fall short in addressing these risks. For instance, authors in [133] propose a presence-based access control system that does not support an extensive set of use cases. Furthermore, other solutions, such as the one in [71], affect the usability of the SPA. As a result, some users trashed their assistants altogether, and companies like Mattel cancelled their assistant projects [105, 177]. Addressing users' concerns about the security and privacy of these assistants is crucial to fostering the trust required to promote their adoption and let users realise their benefits. Most importantly, concerns related to the underlying and integrated technologies that the SPA rely on.

This study aims to develop a more rigorous understanding of SPA's security and privacy risks. In particular, the skills they leverage to expand and extend their functionality and run more complex functions. We analyse skills in the look for concerning privacy practices and study the extent of traceability between the data actions specified in the skill privacy policies and the related data operations obvious to users, where the traceability can be broken, partial or complete depending on how well the data operations and their relationship to the data actions is disclosed. This is paramount for assessing and mitigating vulnerabilities inherent within the SPA, building accountability [57] and implementing adequate defences. This study seeks to address the following research questions:

1. What is the current state of SPA security and privacy?
2. What architectural elements expose SPA to different risks?
3. What data practices used by SPA skills are the most concerning ones?
4. How transparent are the SPA skills developers about these practices?
5. How effective are SPA market operators in helping protect users from malicious skills?

To answer the questions above, we next define the aim of this thesis.

1.2 Research Aims

The goal of this thesis is to investigate the security and privacy risks of the SPA ecosystem and develop methods to provide a more rigorous understanding of them. In particular, the risks associated with the skills and their ecosystem. To achieve this aim, we set out the following research objectives:

- Objective OBJ1 — To produce a systematic state-of-the-art review of the relevant literature on security and privacy issues of SPA and identify key open research directions. The survey should provide a rigorous mapping of the reviewed works to the architectural elements of SPA.
  We accomplish this objective in Chapter 2.

- Objective OBJ2 — To develop a practical method for detecting skills with concerning privacy practices. The proposed technique should consider the complex architecture SPA have and should also support performing this analysis at scale due to the increasingly large amount of skills.
  We meet this objective in Chapter 3 and Chapter 4.

- Objective OBJ3 — To perform a longitudinal measurement study of the SPA skills and explore how the different concerning privacy practices permeate through the markets. The study should identify the key factors that influence these privacy practices and suggest ways to improve them.
  We attain this objective in Chapter 5.

- Objective OBJ4 — To design an open tool that enables proactive audit of SPA skills privacy practices. The tool should be usable for users who do not have a technical background.
  We make progress towards achieving this objective in Chapter 4, and we attain this objective in Chapter 6.

1.3 Thesis Contributions

The main contributions that this thesis brings to the present state of research on the topic can be listed as follows:

- We looked at the generic architecture of SPA and identified the essential points that are important to understand both potential weaknesses and countermeasures.

- We produce a comprehensive review of the existing security and privacy issues, attacks and countermeasures in SPA and present a categorisation for them.

- We conduct a measurement study that provides the first large-scale analysis of the skill ecosystem, analysing over 199k third-party skills. We uncover bad privacy practices in about 43% of the Alexa skills that use permissions involving 50% of the developers with skills that request permissions. The findings led to a responsible disclosure process where we reported 675 Alexa skills with privacy issues to Amazon and all affected developers. As a result, the overall state of privacy in the Alexa ecosystem has improved over time.

- We collect and tag the largest traceability dataset for Alexa known to date, namely the traceability-by-policy dataset (TBPD) and the permission-by-sentence dataset (PBSD) for smart assistants privacy assessment evaluation.

- We train and develop a novel automated traceability analyser that automatically identifies skill traceability. Our system, SkillVet, achieves 93% overall accuracy and, in particular, 99% accuracy for broken traceability. In addition, it can correctly differentiate and classify each of the permissions correctly, obtaining F1 scores and accuracy of over 90% for all data permissions.

- We shed light on how the Alexa ecosystem evolves using data collected across three years between 2019 and 2021. We study developers' data disclosure practices and present an overview of the third-party ecosystem. Through traceability, differential and interrogation analysis, we show that despite the research community continuously contributing to the skill market's sanitation, the skill vetting process still requires significant improvement.

- We implement an online web privacy assessment tool, providing users with an interactive interface that they can use to explore skill traceability regardless of their technical background. The tool can help improve users' awareness of the data practices in SPA to make a better decisions about their data. The tool can also help developers write better privacy policy documents with relevant data practices and assist regulators in simplifying privacy policy documents to detect skills that violate existing regulations.

1.4 List of Publications

The following list of articles have been published/accepted in Journal/conferences in the course of compiling this thesis:

1. **Jide Edu**, Jose Such, and Guillermo Suarez-Tangil. 2020. Smart Home Personal Assistants: A Security and Privacy Review. *ACM Computing Surveys (CSUR)* 53, 6, Article 116 (February 2021), 36 pages. DOI:https://doi.org/10.1145/3412383.

2. **Jide Edu**, Xavier Ferrer-Aran, Jose Such, and Guillermo Suarez-Tangil. 2021. SkillVet: Automated Traceability Analysis of Amazon Alexa Skills. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 15 pages. DOI:https://doi.org/10.1109/TDSC.2021.3129116.

3. **Jide Edu**, Xavier Ferrer-Aran, Jose Such, and Guillermo Suarez-Tangil. 2022. Measuring Alexa Skill Privacy Practices across Three Years. *In Proceedings of the ACM Web Conference 2022 (WWW)*, 11 pages. DOI:https://doi.org/10.1145/3485447.3512289

4. **Jide Edu**, Cliona Mulligan, Fabio Pierazzi, Jason Polakis, Guillermo Suarez-Tangil, and Jose Such. 2022. Exploring the Security and Privacy Risks of Chatbots in Messaging Services. In Internet Measurement Conference (IMC '22), October 25–27, 2022, Nice, France. ACM, New York, NY, USA, 8 pages.

Listed below are other related publications I have been working on:

1. Abdi Noura., Ramokapane Marvin, **Jide Edu**, Jose Such, and Guillermo Suarez-Tangil. 2022. Understanding SPA Skills Developers Security and Privacy Attitudes. Currently under preparation

1.5 Dataset and Code

We make our dataset and code publicly available to support other researchers interested in repeating and reproducing our work. The following are links to the dataset and code generated in the course of compiling this thesis:

1. https://github.com/jideedu/Scrapping-Alexa-Skills

2. https://github.com/jideedu/SkillVet

3. https://github.com/jideedu/Are-We-There-Yet-Alexa-Market-Comparison

4. https://github.com/jideedu/SKILLVET_APP

1.6 Thesis Outline

Each chapter presents an overview of an individual research goal, offers background on the problem, and introduces necessary terminology. The title of each chapter has been chosen to reflect a specific concept to which the thesis address. The contents of the remaining chapters are summarised below:

- Chapter 2 presents background studies relevant to this project on four fronts. The first provides an insight into SPA; the second front offers a classification of the main security and privacy issues of SPA; the third presents a systematic mapping of attacks and countermeasures to the different architectural elements exploiting the vulnerabilities found in SPA; and finally, the last offers a synthesis and summary of the open challenges and suggest future research areas.

- Chapter 3 focuses on third-party skills, which is one of the points we identified in Chapter 2 as needing more attention. Here, we present a systematic study of the SPA skills to understand the type of skills available, their capabilities, how they are being used, and who is behind them. In addition, we study the skill traceability, which lets us understand how many privacy violations there are in the wild of the third-party ecosystem and what is the extent of such violations.

- In Chapter 4 we discuss an automated traceability analyser based on machine learning and natural language processing that automatically identifies the traceability between the permissions requested by the skills and the data practices stated by their privacy policies. We provide a detailed description of the design, implementation and evaluation phase and demonstrate the accurate performance of the system.

- Chapter 5 presents a longitudinal study of the Amazon Alexa skills across the entire marketplace for three years. In particular, we demystify how developers' data disclosure practices have evolved over the years to better understand the various risks the ecosystem presents and aid in formulating appropriate defences for the users.

- Chapter 6 presents the implementation of an online automated web traceability tool that helps identify potential privacy issues in skills. We offer a detailed description of the tool design, architecture, implementation and evaluation.

- Finally, in Chapter 7 we conclude the thesis by providing a summary of the work performed and also suggested directions for future research.

Chapter 2

Literature Review

The aim of this chapter is to fulfil objective OBJ1 presented in Section 1.2: *To produce a state-of-the-art review of the relevant literature on security and privacy issues of SPA and identify key open research directions.* To this end, this chapter presents a review of the background literature upon which the contributions of the thesis are built.

2.1 Introduction

Despite the fast-growing research on SPA's security and privacy issues, the literature lacks a detailed characterisation of these issues. This chapter offers a comprehensive review of existing security and privacy risks, attacks and countermeasures in SPA and presents a categorisation of them. For this, we first provide an overview and background of the architectural elements of SPA, which is vital to understanding both potential weaknesses and countermeasures. In addition, and based on our analysis and categorisation of risks, attacks and countermeasures, we present a roadmap of future research directions in this area. We focus on the following main research questions:

- RQ2.1 — What are the main security and privacy issues behind the use of SPA?
- RQ2.2 — What are the features that characterise the known attacks on SPA?
- RQ2.3 — What are the main limitations of the existing countermeasures, and how can they be improved?
- RQ2.4 — What are the main open challenges to address the security and privacy of SPA?

We used a systematic literature review (SLR) approach [91, 123] to assess existing literature on the security and privacy of SPA. The primary search process involved searching for keywords related to the study (smart home personal assistants, voice assistants, privacy, security) through databases like ACM Digital Library, Web of Science, IEEE Xplore Digital Library, and ScienceDirect. The secondary search process consisted of searching publications manually in the relevant research area for completeness.

Regarding the inclusion and exclusion criteria for the papers, we found through the search process above, we included in this work papers that describe research on SPA or research that is of direct relevance or application to SPA. The papers are reviewed with respect to their techniques, years, criteria, metrics, and results. We exclude position papers or short papers that do not describe any results.

The rest of this chapter is structured as follows: Section 2.2 discusses the SPA architecture and its key components. In Section 2.3, we detail the different security and privacy issues in the SPA. Known attacks on SPA are discussed in Section 2.4. Section 2.5 describes existing countermeasures, and Section 2.6 provides a summary and some discussions on future research directions. Finally, Section 2.8 draws the conclusion.

2.2 SPA Architecture

SPA have a complex architecture (see details in Section 2.2.1). As a general introduction, and despite the fact that different SPA across different vendors have a few distinctive characteristics, all SPA perform similar functions and share some common features. In particular, SPA's architectures usually include, together with other architectural elements such as cloud-based processing and interaction with other smart devices, the following: i) a *voice-based intelligent personal agent* such as Amazon's Alexa, Google's Assistant, Apple's Siri, and Microsoft's Cortana [214]; and ii) a *smart speaker* such as Amazon's Echo family, Microsoft's home speaker, Google's home Speaker, and Apple's HomePod . Note that, while we focus on SPA as one full instantiation and ecosystem based on voice-based personal assistants, some of the issues mentioned in this review may apply to other non-SPA voice-based personal assistants, as there are parts of their architecture that may be similar, especially those parts not related to the smart speaker.

2.2.1 Key Components in the SPA Architecture

SPA are Internet-based systems with a regular iteration of updates. One benefit of this is that its capabilities are wide-ranging and dynamic — they will evolve along with the

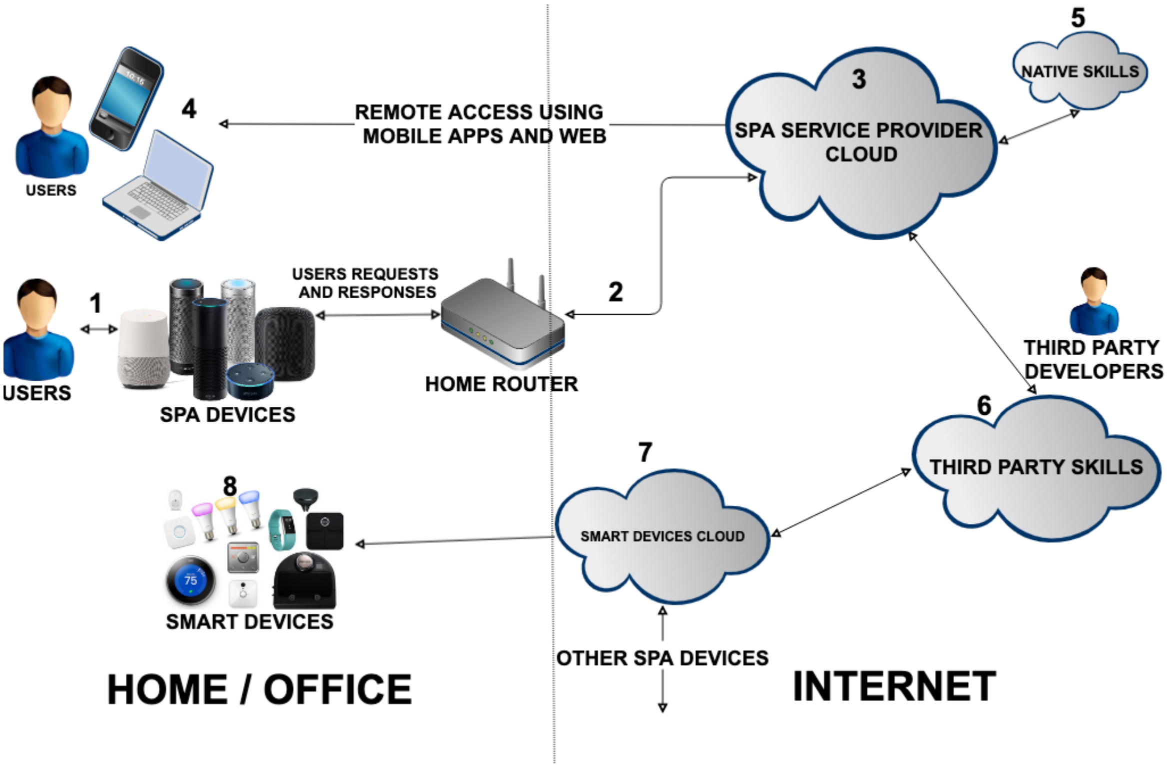


Fig. 2.1 SPA architecture and its key components [63, 21]

proliferation of new Internet services. Figure 2.1 shows the key components in the SPA system architecture. Each component is a potential attack point for an adversary. How some of them may be exploited is discussed in Section 2.4.

Point 1 represents the point of interaction between the users and the SPA devices. SPA devices such as Amazon Echo are equipped with powerful microphones, and the device itself consists of a voice interpreter that records users' utterances. To make use of the SPA, the voice interpreter needs to be activated. Many of the voice interpreters are often pre-activated and run in the background. After the voice interpreter is activated, it then waits for the wake-up word to be triggered [145]. Once it receives the wake-up keyword, it puts the SPA into recording mode. In recording mode, any user utterances are processed and sent through the home router (Point 2) to the SPA cloud (Point 3) [21] for further analysis. Only the wake-up command is executed locally, while all other commands are sent to the cloud. Hence, the SPA must always be online.

The captured utterances are decoded using NLP in the SPA cloud as we detail in Section 2.2.3 below. It must overcome the issue of background noise, echo, and accent variation in the process of extracting the intent [145]. Once the intent is extracted, it is then used to determine which skill to invoke (as discussed in Section 2.2.2). There are two ways to invoke a skill. First, they can be explicitly invoked by using their activation name: for example, where a skill name is “Tutor Head,” it can be triggered by saying “talk to Tutor Head.” Explicit invocation can be extended to use a deep link connection, as detailed here [85] for Google Assistant. For instance, “talk to Tutor Head to find the next course” where the next course is a predefined action under the “Tutor Head” skill. Second, skills can be implicitly invoked by an intent’s query composition without explicitly using their invocation name. If a query does not directly match with a skill, the SPA will either inform the user or match the query to another similar skill when appropriate.

By default, the SPA provider will try to find a native skill to process the request invoked by the user [9]. In this case, the SPA cloud service then sends the intent to its native skill, which processes the request in the cloud of the SPA (Point 5) and sends a response back to the SPA device. When there are no native skills available, the request is sent to third-party skills (Point 6). These are typically hosted in a remote web service host controlled by the developer of the third-party skill. Once the request is processed, the third-party skill returns the answers to the SPA cloud service, which sometimes asks for more information before the request is finalised. In the case where the intent is meant to control other smart devices, the relevant information is forwarded to their respective cloud service (at Point 7), and from there, the instructions are relayed to the target smart device (at Point 8).

2.2.2 Skills

SPA decode users’ voice input using NLP to understand users’ intent. Once the intent is identified, it delegates the requests to a set of *skills* from where it obtains answers and recommendations. Conceptually, skills are similar to mobile apps, which interface with other programs to provide functionality to the user. There are two types of skills, namely: *native skills* and *third-party skills*. The former are skills given by the SPA provider that perform basic functions and leverage providers’ strengths in areas such as productivity (Microsoft Cortana), search (Google Assistant), and e-commerce (Amazon Alexa) [239]. The latter are skills built by third-party developers using *skill kits* [9, 84], which are development frameworks with a set of APIs offered by the SPA provider to perform basic operations.

There are currently thousands of SPA skills hosted online, although the numbers keep growing daily. For example, Amazon’s skill market now has over 100,000 Alexa skills worldwide [117] and the Google Assistant skill market has over 2,000 skills [32].

These skills are classified into different categories such as *home control skills, business and finance skills, health and fitness skills, games and trivia skills, news skills, social skills, sports skills, utilities skills*, etc. As further support to the skills, SPA often have the ability to learn information about users' preferences such as individual language usages, words, searches, and services using Machine Learning (ML) techniques [163] to make them smarter over time.

2.2.3 Natural Language Processing in SPA

SPA benefit from recent advances in Natural Language Processing (NLP), which allow them to handle a wide range of commands and questions. The NLP improvements are attributed to: i) a number of novel advances in ML, ii) a better knowledge of the construction and use of the human language, iii) an increase in the computing power, and iv) the availability of sizeable labelled datasets for training speech engines [94].

Processing user speech includes a complex procedure that involves audio sampling, feature extraction, and speech recognition to transcribe the requests into text. Since humans speak with idioms and acronyms, it takes an extensive natural language analysis to get correct outputs. For instance, issuing a command to SPA asking them to remind you about a meeting at a specific time can be done in several ways. While some parts of this command are more specific than others and can easily be understood, such as the day of the week, other words that support them can be dynamic. This implies that understanding an intention as simple as a meeting reminder might require non-trivial interactions. Figure 2.2 illustrates the process involved in understanding a user's intent and generating responses.

Intent recognition starts with signal processing, which offers the SPA a number of chances to make sense of the audio by cleaning the signal. The idea is to enhance the target signal, which implies recognising the surrounding noise to reduce it [159]. That is one reason why most SPA devices are equipped with multiple microphones to roughly ascertain where the signal is coming from so that the device can concentrate on it. Once the original signal is identified, acoustic echo cancellation [245] is then used to subtract the noise from the received signal so that only the vital signal remains.

Typically, most speech recognition systems work by converting the sound waves from the user's utterances into digital information [79]. This is further analysed in order to extract features from the user's speech, such as frequency and pitch. Primarily, Automatic Speech Recognition (ASR) consists of two steps: features extraction and pattern classifiers using ML [138]. There are several feature extraction methods, with

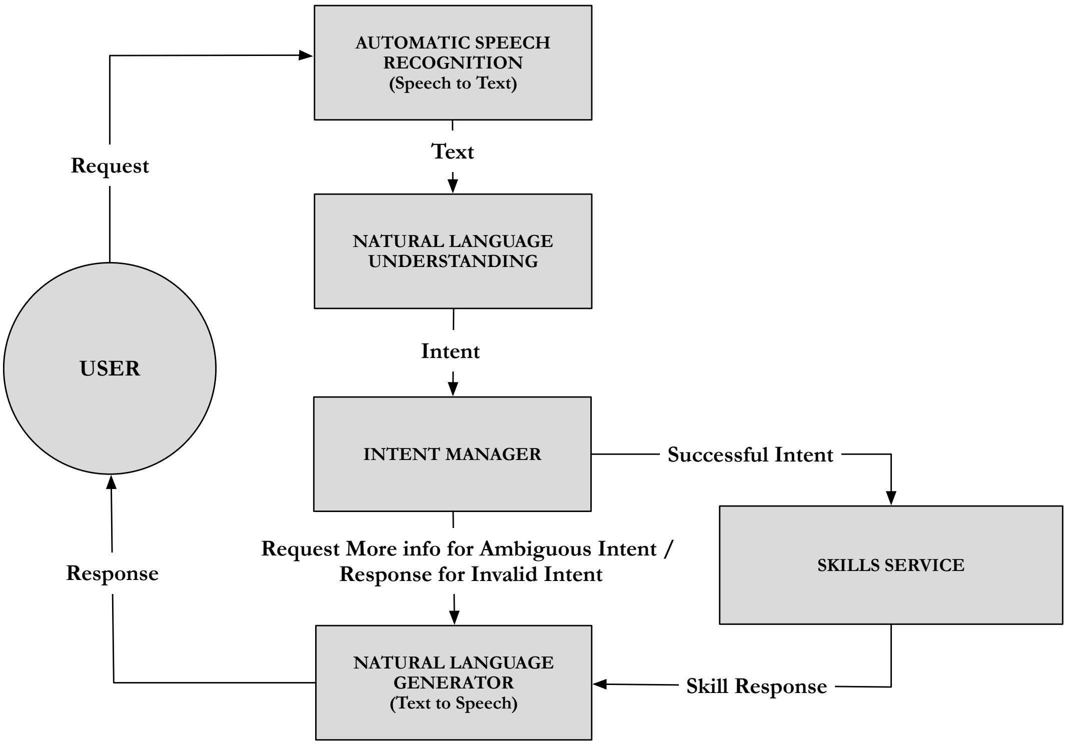

AUTOMATIC SPEECH RECOGNITION (Speech to Text)
Request
Text
NATURAL LANGUAGE UNDERSTANDING
USER
Intent
INTENT MANAGER
Successful Intent
SKILLS SERVICE
Response
Request More info for Ambiguous Intent / Response for Invalid Intent
NATURAL LANGUAGE GENERATOR (Text to Speech)
Skill Response

Fig. 2.2 NLP speech system from *Speech To Text* to *Text To Speech*

Mel frequency cepstral coefficient (MFCC) being one of the most popular since it is believed to mimic the human auditory system [257]. These features are then fed into an acoustic model trained using ML techniques to match the input audio signal to the correct text [175]. For instance, ML models based on Hidden Markov Model (HMM)[79] often compare each part of the waveform against what comes previously and what comes next, and against a dictionary of waveforms to discover what is being said.

Once the SPA cloud has the text that transcribes what the user has said, it employs Natural Language Understanding (NLU), a key component of Natural Language Processing (NLP), to understand what the user intends to do. This is done using discreet to discreet mapping, with some instances relying on statistical models or ML techniques like deep learning to assume the likely intent. The more data available to the NLP system from regular usage, the better the prediction of the user's intent. After the NLU extracts the intent, the intent manager then decides whether more information is needed to provide an accurate answer before forwarding the intent to the skill service for processing. After the intent is processed, the generated skill response is sent to the Natural Language Generation (NLG), where it is converted into natural language representation. It is then communicated back to the user, and it is typically (e.g., Amazon Echo) played by a smart speaker.

2.2.4 Assets in the SPA Architecture

Next, we discuss the assets in the SPA architecture from SPA users' point of view, and why users consider the assets to be important or sensitive, to understand what is at stake, and what should be protected.

2.2.4.1 SPA and Smart Devices

The SPA device and other peripheral smart devices are essential assets in this domain. There are different types of SPA devices attending to where the personal assistant interacts with the user. SPA can be integrated into smart speakers like Amazon Echo, Google Home, and Apple HomePod. As illustrated in Figure 2.1, SPA also interact with other smart home devices [1, 231] such as smart heating and cooling devices (e.g., Nest, or Ecobee 4), Smart security (e.g., Scout, or Abode), smart lighting devices (e.g., Philip Hue, or LIFX), Smart kitchen (e.g., GE+ Geneva) and surveillance cameras (e.g., Cloud Cam, Netgear Arlo Q). All these assets are generally characterised by the hardware they are built on.

2.2.4.2 Personal Data

Personal data is one of the most valuable assets in the SPA ecosystem because of the amount and variety in which personal data is collected, shared, and processed. Therefore, many of the security issues explained below also impact users' privacy, even though this may affect users differently depending on what they value based on their perceptions and preferences [230, 130, 41, 1]. All this may, in turn, be defined by the user's understanding of the data flows in SPA [1] and what they have experienced in other computing contexts [231]. We give more details below of examples of particular types of personal data in the SPA ecosystem.

1. User voice records (audio clips and transcripts): SPA need to continuously learn from past computations for reliable speech recognition. To achieve this, SPA need a large training dataset of user conversations. Users are known to have concerns about the storage of the recordings of those conversations in some cases and, particularly, about what they may be used for [144].

2. User account data: Users also have data as part of their account with the SPA provider. For instance, in Amazon Alexa, this includes users *location, mobile number, email address, name, device address, payment information, and shopping lists [14].* Note, however, this data is not restricted to the SPA provider, and skills

can request permission to access the data from the user's account with the SPA provider.

3. Skill interaction data: Skills can potentially ask users for any personal data through their conversations with the users. In fact, there is research evidence that skills collect personal data during voice interaction without asking for any permissions regarding user account data [165, 88]. According to the research, birthdate, age, and blood type are examples of the data they may ask for.

4. Smart devices data: The integration between SPA and other smart devices brings the smart home into one verbally controlled system and offers the SPA the privilege to manage the services of other connected smart devices. This integration enables access to home sensors that generate valuable personal data.

5. Behavioural data: Apart from the raw data mentioned so far, other sensitive data can be *inferred* from user actions with the SPA or by processing the raw data. This includes predicting users' behavioural characteristics like user interests, usage patterns and sleeping patterns as shown in [46, 47], where the authors demonstrate how personal information can be inferred from data stored by the SPA provider by using a forensic toolkit that extracts valuable artefacts from Amazon Alexa by taking advantage of the Alexa unofficial API.

2.2.4.3 Other Assets

There are other assets such as reputation, financial well-being, physical well-being, emotional well-being, and relationships, all of which could be valued differently by users. For instance, if an attacker successfully breaks into an SPA, users could be affected financially if there are unauthorised purchases, emotionally from shame or embarrassment, as well as suffer damage to their reputation if an adversary uses SPA to impersonate them. In fact, some SPA users restrict the use they make of SPA to avoid impacts on these assets, e.g., many SPA users avoid purchasing through SPA because they do not think the process is secure or trustworthy enough [1].

2.3 Security and Privacy Issues

In this section, we present a classification of the main security and privacy issues of SPA. We use this classification to later map current attacks and countermeasures in Sections 2.4 and 2.5.

2.3.1 Weak Authentication

Here, we discuss issues related to how SPA verify users and how an adversary can exploit such a process.

2.3.1.1 Wake-up Words

By design, SPA authentication is done using wake-up words that are recognised locally in the device. A user has the option to select a wake-up word from a set of predefined options, having one by default. It is therefore very easy for an attacker to infer the wake-up word of the user. In addition to the wake-up word, SPA have no additional ways of authenticating the user. The device will accept any command succeeding the wake-up keyword. Hence, it is easy for anyone in proximity to issue commands to the SPA. Authors in [133, 4, 251, 252] have shown how this weak authentication can be used as a proxy to more elaborated security and privacy attacks. Moreover, SPA can also be activated by conversations that do not contain wake word [66].

2.3.1.2 Always On, Always Listening

As mentioned, the voice command interpreter constantly listens to the user's utterances while waiting for the wake-up word. Having a device permanently on and always listening poses important security and privacy concerns. Accidentally saying the wake-up word or any other phonetically similar words will put the assistants to record. Consequently, any conversation that follows is uploaded to the Internet. This issue could affect the users' privacy in a situation where private or confidential conversations are accidentally leaked or where an attacker can retrieve sensitive information from these devices. Likewise, it could also affect device security as an adversary can issue an unauthorised command to compromise such devices and use them to target other connected smart devices. Recently, due to this feature, a private conversation of a couple was accidentally recorded and sent to a random contact with the Echo device [95]. This example shows that the users are not in total control of their voice data.

2.3.1.3 Synthesized Speech

SPAs are known to listen to audio playback. Just recently, a Tv commercial by Burger King prompted Google Home to read information to the user from Wikipedia about the Whopper hamburger [241]. However, while major SPAs like Alexa and Google have now figured out how to filter out background media [172, 192], they are still vulnerable to synthesising audio that exploits side channels or adversarial examples. For instance, they are vulnerable to inaudible sound reproduced at ultrasonic frequencies [251, 196], and synthesised speech transmitted through electromagnetic radiation. In particular, laser-powered "light commands" [225]. Since the SPA wake-up word can be readily guessed, and the SPA have no means of detecting if a user is in close proximity, there is little or no limit to which speech can be supplied to them and by whom, provided it is meaningful and can be matched with an intent. Synthesised speech (like ultrasonic/inaudible attacks) could offer an adversary a covert channel to issue a malicious command. An attacker could even distribute these speeches over channels like TV and radio to attack multiple targets at once.

2.3.2 Weak Authorisation

In this part, we evaluate the issues regarding how the SPA manage the level of access to data, and the mechanisms users have to control that.

2.3.2.1 Multi-user Environment

The absence of proper functional role separation prevents users from correctly defining what and how resources should be accessed. It is challenging to specify who has access to which resources and how such access should be granted. By default, in a multiuser environment — which many households are, any user can put the SPA into recording mode and issue out instructions to it. Even though the primary user can specify certain access controls for secondary users, the level of granularity is generally coarse and not extensive. For instance, any member of an Amazon household (a feature that allows sharing of contents with family members) can modify the device set-up such as the network connection, sound, and many more without the primary user's consent.

2.3.2.2 Weak Payment Authorisation

SPA systems are increasingly supporting online ordering. Implementing proper security controls challenges usability. For instance, Amazon Alexa users have the option to set a 4-digit PIN code to confirm purchases. At the time of writing, this option is not enabled by default. Even when such an option is turned on, it is vulnerable due to weak lockout[1] implementation [89]. This is because Alexa allows two PIN tries before an ordering process lockout, after which the user has to restart the ordering process

[1] Lockout is a security mechanism that locks an application for some time before a reattempt is allowed.

from the beginning. However, there is no restriction on how many times a user can try to order after every lockout [89]. Following this, vendors have tried to implement alternative countermeasures against misuse in the ordering process. We next show two cases of this. First, some vendors have prevented changes to the shipping address during ordering. However, preventing any change to the shipping address during this process is not enough when dealing with “insiders” (i.e., unauthorised users who have access to the premises where the SPA are installed). The case described in [128] shows how a kid recently made an unauthorised order worth about $300 using her mother’s Amazon account [128]. Second, other vendors have tackled this weak authorisation problem by providing prompt notification to the users about orders. This poses a problem to users who do not frequently check their phones or emails or who may not understand what is happening.

2.3.2.3 External Party

One important concern is how SPA providers, skills developers, developers of integrated smart home devices, and those that have direct access to any of the points of the SPA architecture secure users from external parties that do not have access to any of these points. Like in every other cloud service, the question remains on how data gathered by those involved in the SPA system is shared with third-parties, particularly regarding what kind of controls and mechanisms can be implemented to provide more control to users. Informed decisions can sometimes be taken when third-parties provide privacy policies and terms of use [238]. However, it is currently uncertain what the scope of those terms might imply and how they are enforced.

2.3.3 Profiling

Beyond authorisation, i.e., deciding who has access to what data, there is also the problem of data inference — traditionally known as information processing [211]. Data inference has a particularly dangerous incarnation in SPA in the form of profiling. Profiling identifies, infers, and derives relevant personal information from data collected from users. Profiled data can be related to the interests, behaviours, and preferences of the targeted users [61]. In this subsection, we look into how SPA data can be used to profile users.

2.3.3.1 Traffic Analysis

A good instance of an en-route type of profiling is traffic analysis. An attacker can take advantage of SPA traffic's improper concealment to profile a user as shown in [30]. In particular, attackers can leverage en-route profiling to infer a user's presence. This can be further used to conduct more sophisticated attacks. En-route profiling attacks can be made even when the network traffic is encrypted. While there are obfuscation techniques that can be used to hinder these types of attacks, they have not been adopted in SPA. In this scenario, the most plausible adversary would be a dishonest or unethical Internet service provider. Governments or other global adversaries with access to the user network traffic can also exploit this weakness. The practicality of this threat to encrypted SPA traffic is shown in [30, 198]. While authors in [30] perform traffic analysis without even needing an in-depth inspection of the network packages, MiTM techniques — such as SSL-stripping [253] — might be used to perform profiling over plain-text.

2.3.3.2 Uncontrolled Inferences

Profiling, in this case, is about inferences made by any of the parties in the SPA ecosystem (third-party skill developers, SPA providers, etc.) from data they collect with the consent of the user. This includes some of the personal data mentioned in Section 2.2.4 (conversations, account data, interaction data, etc.). That is, the starting point is data about the user that the user may have consented to share. This data is then used to infer *new* data about the user that the user had not shared. An example would be the behavioural data mentioned in Section 2.2.4. Therefore, the problem is that even when users can choose whether they share some data, they have no control over what the parties can do with the data, or what kind of inferences or aggregations they could make to derive other new personal information about the user, e.g., users' tastes or preferences.

Note that in some cases, collusion between the parties might be possible to be able to conduct more powerful inferences. For instance, malicious skills may collude to aggregate personal data from multiple skills similar to what we have seen in smartphone apps [151]. Here, skill connection pairing [118] may be leveraged to create colluding skills aiming at getting more elaborated profiling. Uncontrolled inferences are especially critical as advances in data analysis enable automated techniques to make sense of unstructured data at scale.

2.3.4 Adversarial AI

As described in Section 2.2.3, for SPA to fulfil the user's request, they needs to first understand what the user's said, understand what the user wants, and before selecting the best skill to fulfil the request. For these, the speech recognition system uses AI techniques like NLP and ML. However, these techniques can introduce the issues discussed below.

2.3.4.1 Adversarial ML

ML in the SPA system is used for many tasks, including speech recognition. Conventionally, ML is designed based on the notion that the environment is safe, and there is no interference during training and testing of the model [173]. However, such an assumption indirectly overlooks cases where adversaries are actively meddling with the learning process [173]. ML is known to be vulnerable to specially-crafted inputs, described as adversarial examples, which are usually derived by slightly modifying legitimate inputs [228]. These perturbations typically remain unknown to the person supervising the ML task but are wrongly classified by already trained ML models. Examples can be used to manipulate what the SPA system understands from spoken user commands [251]. This could then be used to generate a denial of service attack, invoke an incorrect skill [233], or to reduce the ML model quality and performance [40].

Most ML models that perform the same task tend to be affected by similar adversarial inputs even if they use different architectures and are trained on different datasets [174]. This allows the attacker to easily craft adversarial inputs with little knowledge about the target ML model. Research has also shown that speech recognition models often find it challenging to differentiate words with similar phonemes [127], e.g., distinguish between “Cat”, “Pat”, and Fat, which can come in handy when crafting adversarial inputs. Commonly exploited ML vulnerabilities are not the only type of examples that may apply. For instance, to predict the best skill to process the user's request, most SPA continuously learn from the user interactions and regularly retrain their ML models with new data. Attackers could insert adversarial samples into the training dataset to corrupt the ML models (poisoning attack). Another example would be targeting the ML models to extract valuable information (membership inference attack), e.g., the accent of the speakers in speech recognition models [209].

2.3.4.2 NLP Vulnerabilities

Although adversarial ML has a direct effect on the NLP system in SPA as it underpins many NLP tasks used for speech recognition, there are also other parts of the NLP system in SPA that do not directly use ML, but that may also be exploited. Following the example of skill invocation given in the previous subsection, the adversarial NLP problem appears once user utterances have already been transcribed into text and the system needs to decide which skill to invoke from the text (note the difference with the problem of translating into text two words with similar pronunciation).

In particular, Amazon's Echo and Alexa seem to use the lengthiest string match when deciding which skill is called [252]. For example, the text "talk to *tutor head* for me please" will trigger the skill "*tutor head for me*" rather than the skill "*tutor head*." In a similar way to adversarial ML, an attacker could use such difficulty to trick users into invoking a malicious skill intentionally. This can be achieved by registering a skill with the same name (but longest possible string match) than a legitimate skill. Besides, there is currently no restriction on the number of skills that can be registered, hence, an adversary can register as many skills as possible to increase the possibility of getting their skills called.

2.3.5 Underlying and Integrated Technologies

To broaden SPA capabilities and offer ubiquitous services, SPA rely on skills and other existing infrastructures like cloud services and smart devices. This means they can potentially inherit or be subject to issues and vulnerabilities present in or arising from these technologies.

2.3.5.1 Third-party Skills

An attacker could take advantage of lax enforcement of the skill implementation policies and exploit the interaction between the user and the SPA system. For example, by faking the hand over process, a malicious skill can pretend to hand over control to another skill and deceive users into thinking that they are interacting with a different skill (Voice Masquerading attack) in order to eavesdrop on user conversations and collect sensitive information. After all, it is difficult for the user to determine if they are taking to the right skill at a particular period of time because of the vagueness of voice command [165]. Likewise, a malicious skill can fake or ignore the skill termination command and continue to operate stealthily [212]. Furthermore, the existing SPA architecture supports only permission-based access control on sensitive data. It is insufficient at controlling how skills use data once they get access [104]. This could create privacy concerns, especially in over-privileged skills, as it does not allow users to specify the intended data flow patterns once a skill has permission to access data.

In fact, authorising a malicious skill to access confidential information may result in leaking sensitive information to unwanted parties. In the SPA ecosystem, the end-user does not have any kind of access to the skills, which is rather different from the apps in smartphones that will be running in your phone, so protection mechanisms in the smartphone can be used to target apps. In contrast, users don't have a way to install any protection mechanisms beyond those the SPA provider can put in place for skills. A user must rely on the SPA provider to ensure that such services are as secure as they need to be. However, even if the SPA provider would provide a vetting process, related works have shown that they can be successfully evaded [252, 212]. More importantly, a malicious third-party skill could covertly reword responses from legitimate sources to introduce misperception about the reported events intentionally [206].

2.3.5.2 Smart Home Devices

While SPA integration with other smart home devices brings the smart home into one verbally controlled system, it also creates a single key point of interest to attackers. Attackers can take advantage of this in two ways. On the one hand, breaching the SPA can allow attackers to control a wide range of connected devices. More so, privacy issues could emerge from data accumulation, data acquisition, and integration as discussed in [140, 193, 37], where the authors perform a comprehensive review of privacy threats of information linkage from data integration in IoT ecosystems. On the other hand, vulnerabilities in connected smart devices could be used as an intermediate step to attack the SPA [195, 62, 217].

Attacks in connected smart home devices have been investigated in numerous works, including: 1) snooping attack where an adversary listens to the smart home traffic to read confidential data [62], 2) privilege escalation where attackers use design and configuration flaws in smart home devices to elevate privileges and access confidential home users information, 3) insecure interactions between apps that are used for controlling peripheral devices and third-party counterpart apps which could open channels for remote attackers, and 4) other direct compromises of various smart home devices [62, 72]. For instance, the API service on Google Home before mid-July 2018 was reported to be vulnerable to DNS rebinding attacks, which allow remote attackers to initiate a denial of service attack, extract information about the Wi-Fi network or accurately locate this device [166].

It is important to note that some of the issues we identify in this review are not specific to SPA alone. They are also present in other smart home and IoT devices, since the SPA and other IoT devices conduct information exchange and communications in a similar way, and are often co-located within the same environment. Nonetheless, the SPA ecosystem is quite unique, e.g., the speech and intent recognition steps, which determine the actual third-party skill that is to serve a user command, may lead to specific adversarial AI issues as mentioned above.

2.3.5.3 Cloud

While the cloud offers the advantage of having readily available and virtually unlimited resources, it also presents attackers with new opportunities [157]. On the one hand, they are data-rich environments that are centrally located in a single point, and in particular, in SPA architectures, they keep most of the personal data mentioned in Section 2.2.4. Therefore, if this element is breached, attackers may get access to valuable and sensitive information [36]. This is the most concrete and frequently mentioned threat by users regarding smart home data [231].

On the other hand, they usually offer multiple remote ways of accessing the data (e.g., web or app-enabled access) and facilitate online configuration, thereby widening the attack surface. The SPA provider cloud (point #3 in Figure 2.1) is therefore subject to these issues. Most importantly, data in the cloud are subjected to insider attacks (i.e., abuse of authorised access) [181, 215]. For instance, some SPA providers may let employees listen to recorded conversations as they view this process as a critical part of evaluating their SPA speech recognition system [215] and a way of improving customer experience [181]. This is a critical issue when their privacy statements fail to mention this type of usage or whether conversations are used anonymously [191].

Likewise, the SPA provider cloud could also suffer from incomplete data deletion [187]. This situation may enable SPA providers to retain (intentionally or accidentally) private data even after being deleted (assuming users manage to find a way to delete information from the cloud, which is not always easy for them [188]). For instance, it is known that Amazon could keep transcripts of users' voice interactions with Alexa even after the recordings are deleted [114].

2.4 Attacks

This section offers a review of known attacks on the SPA system and examines the vulnerabilities they exploit w.r.t. the issues described in Section 2.3 and the point they target in the architecture in Section 2.2.

Table 2.1 Categorisation of attacks found in previous studies based on vulnerabilities exploited and attack point

Attack Class	Studies	Weak Authentication			Weak Authorization			Profiling		Adversarial AI		Integrated Techs			Attack Points
		Wakeup Word	Always Listening	Synthesized Speech	Payment Auth.	Multiuser Environ.	External Party	Traffic Analysis	Uncont. infer.	Adv ML	NLP Vul	Skills	Cloud	Smart Devices	
Side Channel	Lei Xinyu et al. [133]	✓	✓		✓									✓	1
	Zhang et al. [251]	✓	✓	✓											1
	Segawara et al. [225]	✓	✓	✓	✓										1
	Roy et al. [196]	✓	✓	✓											1
Behavioral Profiling	Apthorpe et. al. [30]							✓							2
Attacks on Voice Models using Adversarial samples	Gong & Poellabaeur[81]	✓	✓							✓					1, 3
	Schönherr et al. [201]	✓	✓							✓					1, 3
	Carlini and Wagner [40]	✓	✓							✓					1, 3
	Vaidya et al. [233]	✓	✓	✓						✓					3
	Carlini et al. [38]	✓	✓	✓						✓					3
Skill Squatting & Masquerading	Zhang et al. [252]									✓	✓	✓			3, 6
	Kumar et al. [127]									✓	✓	✓			3, 6
	Security Research Labs [212]											✓			3, 6

Table 2.1 shows an overview of the most relevant attacks mapped to the vulnerability(ies) they exploit and the affected points in the architecture. We found that most of the attacks target the following elements of the architecture depicted in Figure 2.1:

1. User to SPA device (#1): There is a wide range of attacks targeting this point of the architecture. In particular, we identify related works i) exploiting weak authentication and ii) attacking underlying and integrated technologies.

2. SPA device to SPA service provider cloud (#2): There is an attack reported in the literature that targets this point of the architecture and exploits improper concealment of SPA traffic.

3. SPA service provider cloud (#3): Several attacks are also found at this point of the architecture targeting the SPA cloud components. We identify works exploiting i) ML Vulnerabilities and ii) underlying technologies.

4. Third-party Web skills (#6): Attacks targeting this point of the architecture exploit user misconceptions about the SPA system, and in particular about the skill. We show related works exploiting NLP subsystem vulnerabilities.

We could not find any attacks targeting architectural elements #4 (remote access via mobile and Web), #5 (native Web skills), #7 (smart device cloud), and #8 (connected smart devices). However, this does not mean that attacks targeting those architectural elements are not possible. In fact, some of the threats outlined in [62] and the attacks demonstrated by researchers in [185] could possibly exploit #8. Besides, some of the vulnerabilities that exist in #3 might also be found in #7 as they are both cloud technology. Likewise, attacks targeting #6, such as voice squatting and voice masquerading [252], might also be possible in #5 since both are skill services. Nevertheless, they have not been exploited yet, as far as we know. We discuss this more in detail later on in Section 2.6.

We next describe the attacks we found in related literature by types (or categories) of attacks, particularly looking at the vulnerabilities (described in Section 2.3) that they exploit and the assumptions they make on the environment.

2.4.1 Side Channel Attacks

This includes attacks that are based on information gained from the way the SPA is implemented rather than vulnerabilities in the SPA itself. The *always on, always* *listening* and the *lack of arbitrary wake-up words* within the *weak authentication* category are the most exploited vulnerabilities in this class of attack.

Lei Xinyu et al. [133] look at issues in single-factor authentication methods based on a wake-up word and the lack of a mechanism that can be used to figure out if a user is close-by or not. Using Amazon's Echo device, the authors perform a home burglary attack to manipulate a connected door lock. Likewise, they successfully make a purchase using the compromised device. Authors in [225] also exploit the lack of proper user authentication and vulnerable microphones to inject voice commands into SPA. By simply modulating the amplitude of laser light, the authors successfully use light-injected voice commands to unlock a connected smart lock integrated with the SPA, and to locate, unlock, and start cars (including Ford and Tesla) provided they are linked with the target's Google account. However, unlike in other classes of attacks where attackers are restricted by distance due to the use of sound for signal injection, attackers here are only limited by their capabilities to carefully aim the laser beam on the devices' microphones. Additionally, since light does not penetrate through an opaque object, this attack requires a line of sight to the targeted SPA devices.

The non-linearity in the Micro-Electro-Mechanical Systems (MEMS) microphone over ultrasound is exploited by Zhang et al. [251]. Non-linearity is described as hardware features that cause signals with high-frequency triggers at high power to be shifted to low frequencies by microphones (and speakers) [196]. Even though microphones are designed to be a linear system, they exhibit non-linearity in higher frequencies. By synthesising high-frequency sounds that are not within the human hearing range but are still intelligible to SPA devices, the authors are able to activate, control and issue commands to SPA. This technique is called the dolphin attack as it uses ultrasonic frequencies like what Dolphins use to communicate among themselves. This attack was confirmed on seven popular voice intelligent assistants (Siri, Cortana, Huawei Hi Voice, Google Now, Samsung S Voice, and Alexa) over a range of different voice platforms. On the downside, this attack cannot be conducted above a distance of 5ft from the targeted device. Likewise, it requires specialised hardware to synthesise and play the ultrasonic signal, making it unrealistic for a real-world attack.

In a different study, Roy et al. [196] develop a long-range version of the dolphin attack. They achieved a range of 25ft from their target. By exploiting the non-linearity inside the microphone, like in [251], they generated long-range high-frequency signals that are inaudible to humans but intelligible to SPA. As in the previous study, they control and issue commands to SPA devices with the assumption that the adversary can synthesise a legitimate voice signal. However, rather than using a single ultrasound speaker as done in [251] to play the synthesised signal, the authors used multiple speakers that are physically separated in space. They employ spectrum splicing to optimally slice voice command frequencies and play each slice on independent speakers in a way that the total speaker output is inaudible. Nevertheless, the attack is only feasible in an open environment. This is because high frequencies are more susceptible to interference, which is a limiting factor to the distance [100]. Likewise, this attack requires multiple ultrasound speakers, making it more challenging to implement in a real-world attack.

2.4.2 Behavioural Profiling

At point #2 of the architecture where SPA devices exchange information with the SPA cloud provider, authors in [30] identify privacy vulnerabilities with SPA by passively analysing encrypted smart home traffic. Their study indicates that encryption alone does not offer all the necessary privacy protection requirements. The authors profile users' interaction with Amazon Echo devices by plotting send/receive rates of the stream even with encrypted traffic. This poses a severe privacy implication to smart home users as an attacker can use this to infer their lifestyle and the best time to conduct an attack undetected, as discussed in Section 2.3.3.1. However, the method used in this study might not apply to a situation where different IoT devices communicate with the same domain because of the difficulty of labelling streams by device type.

2.4.3 Attacks on Voice Models using Adversarial Samples

Here, we discuss attacks on speech recognition and processing system using adversarial inputs. Looking at where data-driven ML models operate, authors in [81] show a new end-to-end scheme that creates adversarial inputs by perturbing the raw waveform of an audio recording. With their end-to-end perturbation scheme, the authors crafted adversarial inputs that mislead the ML model. Note that this is widely used in para-linguistic applications. Their adversarial perturbation has a negligible effect on the audio quality and leads to a vital drop in the efficiency of the state-of-the-art deep neural network approach. On the downside, such an attack needs to be embedded in a legitimate audio signal to make them truly obscure. While this attack was not evaluated on a real SPA, it was successful against para-linguistic tasks which are clearly relevant to SPA. In particular, speaker recognition task for performing voice matching [83, 8] to predict the identity of the speaker.

More recently, Schönherr et al. [201] have proposed an adversarial example based on psychoacoustic hiding to exploit the characteristics of Deep Neural Network (DNN) based ASR systems. The attack extended the initial DNN analysis process by adding a back-propagation step to study the level of freedom of an adversarial perturbation in the input signal. It uses forced alignment to identify the best temporal fitting alignment between the maliciously intended transcription and the benign audio sample. It is also used to reduce the perceptibility of the perturbations. The attack is performed against Kaldi,[2] where it obtained up to 98% success rate with a computational effort for a 10-secs sound file in less than 2-mins. However, like in [81], this attack also needs to be embedded in another audio file, which significantly influences the quality of the adversarial example.

Another study conducted by Carlini and Wagner in [40] proposes an attack on speech recognition systems using Connectionist Temporal Classification (CTC) loss. They demonstrated how a carefully designed loss function could be used to generate a better lower-distortion adversarial input. This attack works with a gradient-descent based optimisation [82] and replaces the loss function with the CTC-loss, which is optimised for time sequences. However, the audio adversarial examples generated when played over-the-air cease to be adversarial, making it unrealistic for a real-world attack.

Similarly, Vaidya et al. [233] perform an attack on speech recognition systems using unintelligible sound. This is done by modifying the Mel-Frequency Cepstral Coefficients (MFCC) — feature of the voice command. The attack is performed in two steps: first, altering the input voice signal through feature extraction with adjusted MFCC parameters, and then regenerating an audio signal by applying a reverse MFCC to the extracted features. When put together, this attack is able to craft a well designed adversarial input. The MFCC values are selected in a way that they can create a distorted audio output with the least sufficient acoustic information. This audio output can still achieve the desired classification outcome and is correctly interpreted by the SPA while unintelligible to human listeners. Although this attack successfully exploits the differences between how computers and humans decode speech, it could, however, be detected if a user is in proximity — provided that they hear unsolicited SPA responses.

The attack presented by Vaidya et al. [233] is extended in the work of Carlini et al. [38], where the authors test the attack effectiveness under a more realistic scenario and craft an adversarial example completely imperceptible to humans by leveraging the knowledge of the target speech recognition system.

[2] A widely adopted open-source toolkit written in C++ which offers a wide range of modern algorithms for ASR.

2.4.4 Skill Squatting and Masquerading Attacks

In this section, we discuss attacks that exploit how skills are invoked and the way skills interact with each other.

Authors in [252] target the interaction between third-party skills and the SPA service. Specifically, they analyse two basic threats in Amazon's Alexa and Google's Assistant SPA services: voice squatting and voice masquerading. Voice squatting allows an attacker to use a malicious skill with the longest matching skill name, similar phonemes, or paraphrased name to hijack the voice command of another skill as described in Section 2.3.4.2. In five randomly sampled vulnerable target skills, the authors successfully "hijacked" the skill name of over 50% of them. The feasibility of this type of attack is high, particularly in SPA, such as Alexa that allows multiple skills with the same invocation name. This attack can be used to damage the reputation of a legitimate skill as any poor service of the malicious skill will be blamed on it.

Equally, in a voice masquerading attack, a malicious skill pretends to invoke another skill or fake termination. Then, the skill keeps recording the user's utterances. This attack could be used to snoop on the conversations of the user. While voice squatting attacks exploit the weaknesses in the skill's invocation method, voice masquerading targets users' misconceptions about how SPA skill-switch services work. With some skills requesting private information, an adversary could use these attacks to obtain sensitive information and cause crucial information leakage to unwanted parties. Voice squatting attack is also shown in the work of Kumar et al. [127]. But unlike what was done in [252], Kumar et al. use the intrinsic errors in NLP algorithms and words that are often misinterpreted to craft malicious skills and exploit the ambiguity in the invocation name method.

2.5 Countermeasures

In mitigating the identified risks and attacks, there have been a number of studies proposing various countermeasures. This section summarises research on countermeasures, highlighting limitations and deficiencies. We give a summary of these in Table 2.2. We mapped the proposed countermeasures to the vulnerabilities discussed in Section 2.3. The current mitigation level in the table (last row of Table 2.2) aims to provide a quick indication of the extent the issues identified have been resolved by the countermeasures proposed by the existing publications analysed to date. In some cases, a combination of countermeasures is enough to address a specific concern, while others will require new countermeasures to address them effectively.

The table also has a column called “Usability Impact” to indicate whether usability is considered or not by the countermeasure. We use the symbol “!” where there is “potential usability impact” such as where users are required to put on extra wearable devices (sacrificing user convenience) [115, 71], or the solution might restrict the SPA capability [56], and “?” for the rest, which means “usability not explicitly considered”, as we did not find enough information in the papers to make any claims (positive or negative) about usability. Finally, we also map these countermeasures to the elements of the architecture depicted in Figure 2.1 to describe the points at which the mitigations would be applied. Most countermeasures map to:

1. User to SPA device (#1): There is a wide range of countermeasures proposed to mitigate attacks at this point of the architecture. In particular, we found many related works mitigating *weak authentication* vulnerabilities.

2. SPA device to SPA service provider cloud (#2): At this point of the infrastructure, we found studies proposing different mitigation techniques to obfuscating traffic between the SPA device and the SPA service provider cloud, to mitigate *en-route* vulnerabilities within the *profiling* category.

3. SPA service provider cloud (#3): Few of the existing countermeasures also focused on the *Adversarial AI* vulnerabilities that are found at this point of the architecture and recommended measures aim to mitigate the risks associated with them.

4. New Architecture: Countermeasures in this category modify to some extent the existing SPA architecture as part of the mitigation and/or mitigate vulnerabilities that cut across multiple points of the infrastructure. We mapped these countermeasures to multiple architecture elements to signal where the mitigation applies or the points that would change as part of an architecture modification.

Table 2.2 Categorisation of countermeasures found in related studies

Class	Studies	Weak Authentication			Weak Authorisation			Profiling		Adversarial AI		Integrated Techs.			Mitigating Point	Usability Impact
		Wakeup Word	Always Listening	Synthesised Speech	Payment Auth.	Multiuser Environ.	External Party	Traffic Analysis	Uncont. Inf.	ML Vul.	NLP Vul.	Skills	Cloud	Smart Devices		
Voice Auth.	Voice Match / Profiles [83, 8]			✓	✓										1	?
	Kepuska and Bohouta [115]	✓	✓	✓											1	!
	Huan et al. [71]	✓	✓	✓	✓										1	!
	Chen et al. [43]			✓											1	?
Location Verification	Lei Xinyu et al. [133]	✓	✓												1	?
Spectral Analysis & Frequency Filtering	Roy et al. [196]			✓											1	?
	Zhang et al. [251]			✓											1	?
	Lavrentyeva et al. [131]			✓											3	?
	Malik et al. [143]			✓											3	?
Traffic Shaping	Liu et al. [137]							✓							2	?
	Park et al. [176]							✓							2	?
	Apthorpe et al. [29]							✓							2	?
Command & Phonetic Analysis	Zhang et al. [252]											✓			3	?
	Kumar et al. [127]										✓	✓			3	?
New Architecture	Coucke et al. [56]	✓					✓	✓	✓			✓	✓		New Arch.	!
	Aloufi et al. [6]						✓		✓				✓		1	?
Current Mitigation Level		◐	◐	●	◐	○	◐	◐	◐	◐	◐	◐	◐	○		

2.5.1 Voice Authentication

One of the defences that have been put in place against weak authentication is voice authentication. With this defence, the SPA can tell apart individual users when they speak. For instance, some SPA such as Google and Amazon perform speaker verification through voice authentication, known as *voice match* [83] and *voice profiles* [8] respectively. However, none of these mechanisms is enabled by default, and it is left to the users first to realise their existence and then decide whether they would like to activate them. Moreover, even when these mechanisms are activated, they are still open to attack as an attacker can still trick the system with a collected or synthesised voice sample of the legitimate user [43]. Collecting voice samples is easy since the human voice is open to the public. Moreover, unlike passwords that can easily be changed if compromised, a human voice is a feature that is difficult to replace.

Another important voice authentication method is proposed in [71]. In this study, the authors present a continuous authentication VAuth system that ensures that the SPA works only on legitimate users' commands. The solution consists of a wearable security token that repeatedly correlates the utterances received by the spa with the body-surface vibrations it acquires from the legitimate user. The solution was said to achieve close to 0.1% false positive and 97% detection accuracy and works regardless of differences in accents, languages, and mobility. However, though this system achieves a high detection accuracy, wearing devices such as eyeglasses, headsets, and necklaces would introduce a potentially unbearable burden and inconvenience to the users.

Kepuska and Bohouta [115] also proposed a multi-modal dialogue system that combines more than one of voice, video, manual gestures, touch, graphics, gaze, and head and body movement for secure SPA authentication. Unfortunately, even though this system might solve the authentication and voice impersonation challenges earlier discussed, the authors have only been able to test the system's individual components and not the entire system as a whole. Finally, Chen et al. [43] propose a software-only impersonation defensive system. The system is developed based on the notion that most synthesised speech needs a loudspeaker to play the sound to an SPA device. As conventional loudspeakers generate a magnetic field when broadcasting a sound, the system monitors the magnetometer reading, which is used to distinguish between voice commands from a human speaker and a loudspeaker. In a situation where the magnetic field emitted is too small to be detected, the system uses the channel size of the sound source to develop a means of authenticating the sound source.

However, the effectiveness of the system depends heavily on the environmental magnetic interference. Likewise, the sound source needs to be at a distance of more than 2.3in (6cm) from the system to prevent the magnetic field from interfering with the magnetometer's reading. In addition, the system has a high false acceptance rate when the sound source distance to their system is greater than 4in (10cm) in a situation where the loudspeaker magnetic field is un-shielded and less than about 3in (8cm) when shielded.

2.5.2 Location Verification

Another important measure implemented against weak authentication is a presence-based access control system. This system allows SPA to verify if a user is truly nearby before accepting any voice commands. Lei Xinyu et al. [133] propose a solution that uses the channel states information of the router Wi-Fi technology to detect human motions. Interestingly, it eliminates the need for wearable devices and introduces no added development cost as it uses the existing home Wi-Fi infrastructure. The solution has an advantage over the traditional voice biometrics recognition, i.e., that becomes ineffective as users age, become tired, or ill. However, the system's effectiveness depends on selecting the best location for the Wi-Fi devices and setting the right parameters for the detection. Besides, it only supports commands that come from the same room where the SPA device is deployed: in their case, an Amazon Echo. Likewise, the system is situational as it works best if there is no structural change to the location where the devices are deployed.

2.5.3 Frequency Filtering & Spectral Analysis

Another category of countermeasures aims to enhance authentication, particularly by protecting the SPA against synthesised speech using frequency filtering and spectral analysis.

In the work of Roy et al. [196], the authors propose a system nicknamed *lip read* that is based on the assumption that some of the features of voice signals–basic frequencies and pitch–is preserved when it passes through non-linearity. It was reported that this system obtains a precision rate of 98% and a recall rate of 99% in a situation where the adversary does not influence the attack command. However, there is no formal guarantee of this countermeasure as they are unable to model the frequency and phase responses for general voice commands. Likewise, their defence only considers inaudible voice attacks ignoring finding the true trace of non-linearity.

Similarly, Zhang et al. [251] propose another set of countermeasures against synthesised speech attacks. The authors recommend two hardware-based mitigating measures — the first one aims to enhance the microphones used by the SPA devices. In contrast, the latter hardware-based defence is intended to cancel any unwanted baseband signal. Enhancing the microphone approach entails designing an improved microphone similar to the one found in Apple iPhone 6 plus that can subdue any ultrasonic sound. On the other hand, cancelling the unwanted baseband signal of the inaudible voice command solution entails introducing a module before the low pass filter in the subsystem used for voice capturing to identify and cancel the inaudible voice commands baseband signal. Likewise, the software-based countermeasure relies on the principle that a demodulated attack signal can be distinguished from legitimate ones using a machine-based learning classifier.

In another study, Malik et al. [143] proposed a countermeasure based on higher-order spectral analysis (HOSA) features to detect replay attacks on SPA. The authors show that replay attacks introduce non-linearity, which can be a parameter to detect it. Lavrentyeva et al. [131] also explore different countermeasures to defend against voice replay attacks. Even though the countermeasure is implemented at #3 of the architecture because it needs extensive computational power, it aims to secure #1.

The researchers use a reduced version of Light Convolutional Neural Network architecture (LCNN) based on the Max-Feature-Map activation (MFM). The LCNN approach with Fast Fourier Transform (FFT) based features obtained an equal error rate of 7.34% on the ASVspoof 2017 dataset compared with the spoofing detection method in [232] with an error rate of 30.74%. The authors further utilised the Support Vector Machine (SVM) classifier to offer valuable input into their system's efficiency. Consequently, their primary system based on systems scores fusion of LCNN (with FFT based features), SVM (i-vector approach), recurrent neural network (RNN), and convolutional neural network (with FFT based features) shows a better equal error rate of 6.73% on their evaluation dataset.

2.5.4 Traffic Shaping

To defend against profiling, Liu et al. [137] propose a countermeasure to mitigate traffic analysis vulnerabilities (part of the *profiling* category). The authors present a solution that protects the smart home against traffic analysis — a community-based "differential privacy framework". The framework route traffic between different gateway routers of multiple cooperating smart homes before sending it to the Internet. This masks the source of the traffic with little bandwidth overhead. Nevertheless, this approach requires cooperation from multiple homes, which makes it challenging to implement. In addition, it could result in long network latency if the homes are not geographically close.

Other approaches can leverage traffic shaping to prevent profiling. For instance, in [176], Park et al. conceal smart home traffic patterns using dummy activities that have a high likelihood of occurrence. This is done considering the behaviour of the inhabitants of that environment during the time of measurement. While this technique is energy efficient and supports low latency transmission of real data, its implementation requires the participation of many devices and can not shape traffic from genuine user activities.

In another study [29], Apthorpe et al. propose a traffic shaping algorithm to make it challenging for an adversary to effectively distinguish dummy traffic patterns generated to mimic genuine user activities from the actual genuine traffic. However, this method only works against a passive network adversary and protects only traffic rate metadata such as packet times and sizes. This approach needs to be used with other methods to protect the categorical metadata such as protocol, IP address, and DNS hostnames. Likewise, the bandwidth overheads required to reduce the adversary confidence varies with respect to the type of device being protected. In fact, most of the existing traffic shaping techniques depend on effectively mimicking and realistic timing fake user activities.

2.5.5 Command and Phonetic Analysis

Here, we discuss countermeasures aiming at mitigating the issues of malicious skills. In particular, the skill vulnerabilities exploiting the interaction between the user and the third-party skill services.

Zhang et al. [252] present a system that examines the skill's response and the user's utterance to detect malicious skills that pretend to hand over control to another skill and deceive users into thinking that they are interacting with a different skill. The system relies on a User Intention Classifier (UIC) and a Skills Response Checker (SRC). The SRC semantically analyses the skill response and compares it against utterances from a black-list of malicious skill responses to flag off any malicious response. While the user UIC, on the other hand, protects the user by checking their utterances to correctly determine their intents of context switches.[3] This is done by matching the meaning of what the user says to the context of the skill the user is presently interacting with and also that of the system commands. They also consider the link between what the user

[3] This is, examining the intents of changing from one task to the other.

says and the skill they are currently using. UIC complements the SRC, and their system reports an overall detection precision rate of 95.60%. Nevertheless, one key shortcoming of this system is the difficulty in implementing a generic UIC due to variation in Natural language-based commands and how to distinguish legitimate commands.

In a similar study, Kumar et al. [127] suggests performing phonetic and text analysis for every new skill's invocation name to mitigate voice squatting attacks. They check whether the new skill's invocation name can be mistaken with an existing one, vetting then the creation of the clashing skill. Their solution is similar to what is currently being implemented during domain registration, where registrars do not register domain names that resemble popular domains.

2.5.6 New Architecture

In this section, we discuss countermeasures that propose a novel architecture for SPA, different from the one described in Section 2.2.1. In particular, we discuss the work proposed by Coucke et al. [56], and Aloufi et al. [6] which propose changes to the architecture, particularly in terms of the speech recognition functionality.

In the work of Aloufi et al. [6], the authors proposed a privacy-preserving intermediate layer between users and cloud services to sanitise voice input directly at edge devices. To ensure privacy protection, the proposed layer collects real-time speech data and uses CycleGAN-based speech conversion to remove sensitive information before forwarding it to service providers. An experimental evaluation to assess the efficacy of the proposed method enables the identification and removal of sensitive, emotional state information by 91%. However, one limitation of the framework is that it only protects users' privacy in conversation data. It does not safeguard privacy in data that are requested from users' accounts or data inferred from users' interactions with the SPA.

Coucke et al. [56] present a *privacy by design spoken Language Understanding platform* that does not send user queries to the cloud for processing. The speech recognition and the intent extraction are done locally on the SPA devices themselves using a partially trained model with *crowd-sourced* data and using *semi-supervised* learning. Many use cases do not need Internet access. However, when the use case requires internet access, such as when data needs to be retrieved or transmitted to an Internet service, then the system processes the data within the SPA device where it was generated rather than in the cloud. This makes it hard for an adversary to perform a mass attack as they can only target a single user or device at once. With such an infrastructure, issues related to *always on always listening*, *cloud*, and *third-party access*, have limited impact since the data is processed locally. Besides, it allows personalising the wake-up word, mitigating the wake-up word vulnerability introduced in Section 2.3.1.1.

However, the platform requires a user to specify the skills on which their assistant will be trained on. Hence, such an assistant can only work within predefined scopes of the selected skills on which their model was trained, thereby restricting their capabilities to only those skills used for their training. It is important also to note that, although this infrastructure modifies the existing SPA architecture so that speech recognition and intent identification is conducted locally, it does not completely eliminate data transmission to other devices or cloud services. The SPA still communicates with other connected devices or cloud services depending on the context of use. This means that attacks like the one described in [195] may still be possible.

2.6 Discussion and Open Challenges

Building on the analysis and categorisation of the related literature studied in the previous sections, we then offer a synthesis and summary of this review and suggest future research areas.

One can easily observe in Table 2.1 that vulnerabilities related to weak authentication are the most exploited flaws. The *wake-up word* and the *always listening features* are typically combined and can be described as the gateway of synthesised speech attacks. No related works currently exploit the multiuser environment and external party access. We also observed that the majority of the attacks target point #1 of the architecture: the point of interaction between the users and the SPA devices as it requires an attacker with lower capabilities. Although few attacks exploit more than one point of the architecture — e.g. [252, 127, 81], none is observed at point #5, point #7 and #8 even though attacks targeting those architectural elements seem possible as discussed in Section 2.4.

Similarly, Table 2.2 shows that countermeasures for *weak authentication* vulnerabilities, and in particular countermeasures towards mitigating synthesised speech have received wide attention in the literature. Taking both Table 2.1 and Table 2.2, we can see a concentration of research efforts towards one particular part of the whole SPA architecture, the direct interaction between the user and the smart speaker — or point #1 of the architecture. While indeed, this is an important part of the architecture, SPA should consider security in a holistic manner. This shows that despite the growing research efforts in security and privacy in SPA, we, as a community, also need to recognise and tackle SPA problems that go beyond that point of the architecture.

Based on our findings, we suggest a number of open challenges in SPA. These include: i) a practical evaluation of existing attacks and countermeasures, ii) making authentication and authorisation stronger as well as smarter, iii) building secure and privacy-aware speech recognition, iv) conducting systematic security and privacy assessments to understand the SPA ecosystem and associated risks better, v) increasing user awareness and the usability of security and privacy mechanisms in SPA, and vi) understanding better profiling risks and potential countermeasures. All of which are discussed below in the following subsections.

2.6.1 Practical Evaluation of Existing Attacks and Countermeasures

We observed that many of the attacks target the underlying hardware of the voice infrastructure. For instance, [196] and [251] use high frequencies signals to attack the non-linearity in SPA devices microphones. While some of these attacks synthesise speech in a way that may be intelligible to humans and easily noticed by users in proximity [133], other attacks synthesise speech in a way that is unintelligible to the users [251, 196]. Thus, one could argue that the second type of attack is more likely to succeed in practice than the first type.

Our study also revealed that many attacks require different domain-specific knowledge to be successful, which might not always be available. For example, attacks conducted in [196, 233, 81] need knowledge of the machine classifiers, while the one demonstrated in [252] requires the understanding of the SPA skills invocation model. In some cases, this knowledge is available or can be reverse-engineered from interactions with the SPA and their architecture. However, beyond these observations that we can derive from a literature review, some important questions remain unanswered, such as:

1. What is the severity of the existing attacks?
2. What is the likelihood of success of these attacks in practice?
3. What is the cost associated with existing attacks and countermeasures?
4. What is the effectiveness of these countermeasures? and
5. How usable are these countermeasures?

2.6.2 Making Authentication Stronger

Despite receiving most of the attention in terms of countermeasures, with some of the issues and attacks having a counterpart countermeasure, weak authentication issues have not yet been completely addressed. As discussed earlier, many of the attacks targeting the SPA system exploit its weak authentication, especially the *always on, always listening features*. This attack is usually combined with other vulnerabilities. Although one could say that the *always on, always listening features* improve the responsiveness of the devices by making resources available to the user before they start uttering commands, the security and privacy risks may outweigh the benefit. Several independent input variables such as voice, video, manual gestures, touch, graphics, gaze, and others like the solution proposed in [115] could be combined to make authentication stronger.

However, most SPA are designed without environmental sensors. The lack of environmental sensors makes it challenging to implement context-aware authentication systems that could sense the physical environment, and leverage such information to adjust the security parameters accordingly. Also, there may be privacy issues and concerns when using even more personal information (e.g., video). Likewise, current authentication mechanisms in integrated technologies like other smart home devices are decentralised. Each integrated technology has its own authentication mechanism. By implementing a centralised mechanism, potentially in an SPA, a user could access multiple integrated technologies by authenticating only once. This would enhance usability by lessening the authentication burden on users and improving security as it would ensure consistent authentication across smart home devices. However, this needs to be implemented carefully so as not to create a single point of failure.

Future research can also consider how communication protocols may improve current authentication mechanisms in SPA. There are examples of how these mechanisms can be used in other systems such as remote car unlocking and contactless payment, where they are becoming an effective way to verify users' presence [33]. Popular among them are the distance-bounding protocols, which can be used to authenticate the user and access their location. These protocols have proven to be practicable, especially in a system that is susceptible to distance-based frauds. Distance-bounding protocols are based on timing the delay between when a verifier sends a challenge to the moment the response is received. This allows the verifier to detect a third-party interference as any sudden delay in the proper response, which is considered to be the result of a delay due to long-distance transmissions [236, 33, 156]. Nevertheless, the effectiveness of this protocol depends on getting the correct propagation time.

2.6.3 Enhanced Authorization Models and Mechanisms

More flexible access control and authorisation models and mechanisms are needed. These mechanisms should be able to dynamically authorise and adapt permissions to users based on the current context and their preferences.

According to a recent study, users preferred authorisation policies in smart homes are affected by some distinct factors [93]: i) the capabilities within a single device, ii) who is trying to use that capability, iii) and the context of use. Hence, designing authorisation models that consider SPA capabilities and the context of use may help create authorisation rules that adequately balance security, privacy, and functionality. In fact, similar models have already been implemented successfully in other domains like smartphones [167]. Furthermore, we have observed that SPA requires more fine-grained authorisation mechanisms. This not only applies to the voice of the user itself, but also to the data that can be obtained from how users interact with the devices. In particular, these interactions can be used to infer, for instance, a user's sleeping patterns, as discussed earlier.

Novel authorisation models and mechanisms for SPA should consider not only single users but also multiple users. However, there are no security and privacy mechanisms for SPA that considers *multiuser environment* issues. This is important, as even if SPA would support multiple accounts, it is a common practice to share accounts between multiple users [148] (especially if one of the accounts has more privileges). The lack of proper authorisation can prompt insider misuse, e.g., members of the household spying on their partners [76], which can be particularly problematic in the case of intimate partner abuse [149]. Moreover, smart home data is relational, and it usually refers to a group of people collectively [178], e.g., if there is a way to infer whether there is someone at home or not, this already gives information that can be sensitive to everyone living there. Some general-purpose smart home privacy-enhancing IoT infrastructures like the Databox [178, 7] recognise the multiuser problem but no solution has been proposed yet in general for smart homes or in particular for multiuser sharing management in SPA. A great deal of research on methods and tools to help users manage data sharing in multiuser and multiparty scenarios have been proposed for social media (see [221] for a survey), and particular methods for detecting and resolving multiuser data sharing conflicts, such as [220], could be adapted from there or used to inspire multiuser solutions for the SPA case.

Furthermore, the existing SPA architecture supports only permission-based access control on sensitive data, which is insufficient for controlling how third-party skills use data once they get access. There is a need for research to should study how to implement a framework that allows users to pronounce their intended data flow patterns. Similar frameworks [73, 104] have been successfully applied in smartphones for IoT apps. Also, there is a lack of authorisation frameworks for data generated during user interactions with a third-party skill, which is one of the personal data assets mentioned in Section 2.2.4. Novel authorisation mechanisms that allow users to specify, monitor and control what data can be shared with those that have no direct access to the SPA architecture, under what condition should the data be shared (reason), how it should be shared (means) and what it can be used for (purpose) could also help address the issue of external parties.

2.6.4 Secure and Privacy Aware Speech Recognition

NLP and ML models are used in conjunction for speech recognition. Therefore, protecting these models against manipulation, e.g., through well-crafted adversarial inputs as pointed out in Section 2.3.4, becomes paramount.

It is apparent from Table 2.1 and Table 2.2 above that many attacks are exploiting adversarial ML and NLP issues, and there are substantially more attacks than defences studied in the related literature. Therefore, SPA providers need to consider adversarial examples when developing their speech recognition models. However, that is not an easy task, and more research is required in this direction. Some existing countermeasures used in other domains, such as adversarial training and distillation, could help to develop robust ML models for speech recognition in SPA, but they can be defeated using black-box attacks or attacks that are constructed on iterative optimisation [39]. Also, validating the input and reprocessing it to eliminate possible adversarial manipulations before it is fed to the model is a countermeasure that greatly depends on the domain and is subjected to environmental factors [173]. Likewise, testing is not enough to secure ML, as an adversary can use a different input from those used for the testing process [82].

Furthermore, the performance of the current speech recognition system still deserves improvement as shown earlier — recall that these systems often find it difficult to i) understand words with similar phonemes [127], ii) understand different but similar words, and iii) resolve variation in natural language-based command words [251]. Since the word error rate (WER) is the common metric used for evaluating the performance of automatic speech recognition systems [50], it may be easy for an adversary to craft an adversarial input that could maximise the WER of the speech recognition system by exploiting the NLP framework and the ML techniques. This is shown in [251], where the speech recognition system is exploited to manipulate the intent that the system understands from the user's command.

Beyond security, obtaining valuable information from big data while still protecting users' privacy has become interesting research in data analysis. While SPA providers let users review and delete their voice recordings, a recent study shows that users are unaware (or do not use) those privacy controls [130]. It is also unclear how effective these controls actually are even if used, e.g., these controls allow the user to delete particular raw utterances but they cannot delete what could be inferred from them (i.e., the model) [114]. In light of this, SPA vendors need to understand the privacy challenges of machine learning. For instance, although most existing SPA providers aim to ensure privacy while processing users' voices in the cloud, that is a challenging endeavour with current SPA architectures. With edge computing gradually coming into the limelight, data can now be processed locally, where it is generated, rather than being transmitted to a centralised data processing centre [194, 254]. This helps reduce the current dependency on the Internet and eliminates the necessity of putting sensitive data into the cloud.

While related work [56] addresses this direction with a decentralised voice processing platform, it is challenging to build a general-purpose SPA using such platforms. This is because SPA developed with such platforms can only work within predefined scopes of the selected skills on which their model was trained. Therefore, there is a need for future efforts on how to make voice processing privacy-preserving without hindering SPA's capabilities effectively.

2.6.5 AI-based Security and Privacy

In addition to using AI techniques for SPA functionality, e.g., speech recognition, they could also be used to make SPA more secure and aid users in managing their privacy as they see fit. AI techniques would include not only data-driven techniques like ML but also knowledge-based techniques such as normative systems and argumentation, which have been successfully used to develop intelligent security and privacy methods in other domains [222, 218]. AI techniques could be used to address the issue of *always on always listening* and*synthesised speech* under the weak authentication vulnerabilities. For instance, it could be applied to detect malicious commands being spoken to the SPA devices (i.e., to make authentication stronger and more resilient to attacks). Likewise, it could be used to solve the issue of *multiuser authorisation and over-privileged skills* by applying it to help primary users configure the permissions they grant to other users and third-parties skills, respectively. Similar research has already been shown to detect intrusions [54] and to help users in other domains like mobile App permission management [169] and Social Media privacy settings and data sharing [153, 154]. As for speech recognition, these ML-based methods need to be engineered considering adversarial cases [82].

Examples of the use of knowledge-based AI techniques include the use of norms, which have been widely explored in recent years, especially to reduce the autonomy of autonomous and intelligent systems to conform to decent behaviours [58]. Norms are usually delineated formally using deontic logic to state what is permissible, obligatory, and prohibited, providing a rich framework to express context-dependent policies, e.g., based on Contextual Integrity [168], and they can be defined, verified, and monitored for socio-technical systems like SPA [107, 60]. For instance, norms would be beneficial to avoid issues like the case discussed in [240] where a private conversation is recorded by an Alexa and forwarded to a random contact, as a norm could specify the type of conversations that may or may not be shared with particular contacts, and that norm could be verified and monitored for compliance. Another example is norms that govern multiuser interactions with the SPA as discussed in Section 2.6.3. Norms for SPA could be elicited automatically as in [59] or by crowd-sourcing the acceptable flows of information as in [75].

Another knowledge-based AI technique like automated negotiation [34, 224] could be used to help SPA users navigate the trade-offs and negotiate consent in the very complex SPA ecosystem, including third-party skills and smart devices. For instance, instead of having the user manually inspect and approve every permission for the many third-party skills that may request them (as it happens now in SPA ecosystems like Amazon Alexa and Google Home), the SPA could automatically negotiate those permissions with the third-party skills. This can be done, however, always in a way in which consent could be revocable and access patterns apparent to the user on-demand, allowing reactive and dynamic data sharing adjustment. Finally, other AI techniques like computational trust [182] could be used to choose and only share data with third-party skills and smart devices that are privacy-respecting and trustworthy.

2.6.6 Systematic Security and Privacy Assessments

SPA are a type of cyber-physical system. Previous research looked at how the assurance techniques and testing methodologies most commonly used in conventional IT systems [184] apply to cyber-physical systems, including penetration testing, static &

dynamic analysis, fuzzing, and formal verification. However, it is still unclear how these security testing techniques apply to the SPA system and what are the practices used by third-party developers in this ecosystem.

Assurance techniques are known to have different cost-effectiveness in practice [223], and that cost-effectiveness for one very same assurance technique has been shown to vary across different cyber-physical systems [31], such as Industrial Control Systems [125]. Therefore, a direction for future research is to study and evaluate how these assurance techniques will perform for the case of SPA and whether or not SPA's unique features like voice recognition and its integration with other technologies like the cloud and other smart devices require novel techniques or methodologies. For instance, the known potential to have composite vulnerabilities that exploit both the physical and the IT part of cyber-physical systems [49, 48] has already been shown to also apply to SPA, e.g., [196].

Additionally, authors in [251] show that physical properties can be used to compromise the SPA by using high frequencies signals to attack the non-linearity in SPA devices' microphones as detailed above in Section 2.4.1. A set of key research questions to answer revolve around which assurance techniques can be used to improve security in SPA systems (see Appendix A in [125]). In particular:

1. Can a review of standards and procedures be used to mitigate security risks in SPA systems?

2. Can we run dynamic analysis techniques over components of the SPA architecture? and

3. Can we devise a methodology to provide an independent validation when many components of the SPA system are hosted in the cloud?

Future work should also look at the best and most systematic way to conduct privacy assessments in SPA [243]. However, it remains unclear how many privacy violations there are in the wild of the third-party ecosystem and what is the extent of such violations. Measuring privacy violations systematically is particularly challenging as privacy policies are usually unstructured data. Thus, it is hard to infer properties from them automatically. Of particular interest might be to study the (extent of) traceability between the actions of the data specified in privacy policies, such as those in the privacy policies of the third-party skills developers in SPA, and the related data operations obvious to users via SPA and/or associated smartphone interfaces, which will also be crucial to help tackle the current *weak authorisation* and *profiling* issues of SPA.

One important research question is whether related works could be adopted to measure policy traceability in the SPA domain. Methodologies could be adapted from the social media [28] and smartphone apps [155], which already showed the extent of traceability in these domains, together with methods to help developers automatically map traceability between policies and operationalised controls and maintain it through the development cycle [27]. As real breaches happen (e.g., [240]), methods to study whether there are gaps in security and privacy policies, such as [108] applied to SPA, would also be helpful. Thus, a systematic study could measure how many privacy policies are complete and broken for the third-party SPA ecosystem. This will further ensure a better understanding of the different risks the ecosystem presents and aid in formulating appropriate security and privacy policies for the users.

2.6.7 Increasing User Awareness

Although implementing a technical defensive measure might go a long way in mitigating some of the identified risks, effective countermeasures will be difficult without better user awareness. Research shows that the lack of awareness about data practices in smart home devices affects users' security and privacy practices [231]. Some SPA users are not very concerned when it comes to the security and privacy issues in SPA [249], as they believe they are not valuable targets for attackers [231], or they simply exhibit inaccurate and incomplete mental models of the SPA ecosystem [1]. Therefore, it is essential that users understand the risks and threats present in the SPA ecosystem, including the assets that can be compromised and why they need protection for better risk management.

Users should be well informed to adopt best practices and even understand what key steps they have to take when either their security or privacy is breached [246]. One crucial way of keeping SPA users informed is to design usable privacy notice that helps them understand and manage their data in SPA, accompanied with usable security and privacy mechanisms (as discussed below in Section 2.6.8). Privacy notices must be relevant, actionable and understandable as discussed in [200], and their design should be considered along four main dimensions:

1. Timing — when should a privacy notice be presented;

2. Channel — how should the privacy notice be delivered;

3. Modality — how the information should be conveyed; and

4. Control — how choice options are integrated into the privacy notice.

Another example would be leveraging the already discussed assessments in Section 2.6.6, in order to produce a white (or black) list of third-party skills based on the level of security and/or privacy, they offer considering the results of the assessments.

2.6.8 Usability of SPA Security and Privacy Mechanisms

While users' awareness is crucial in understanding the system's risks, awareness without usable security and privacy controls mechanisms may not be effective in mitigating these risks. For instance, some SPA users, while aware of some risks, do not know how they can protect themselves [1]. In addition to knowing the mechanisms they could use to protect themselves (such as those to achieve a basic level of cyber hygiene [219] but in the SPA domain), users should be able to utilise any SPA security and privacy mechanisms in a convenient manner that does not affect usability or functionality of SPA. This is because convenience and connectivity are important concerns for smart home users, influence their perceptions and opinions, and their attitudes towards external entities that design and regulate SPA [255].

Nonetheless, these measures' primary concern is that they have an important impact on usability, as they clash with the sought "hands-free" experience when interacting with SPA. In some other cases where non-technical coping strategies may not be available, SPA users are merely avoiding the SPA functionality they perceive to be risky, e.g., some SPA users only create shopping lists through the SPA but buy the items using the traditional web interface as they perceive buying through the SPA as risky and do not know how to protect themselves [1].

From all the technical countermeasures that we surveyed (see Section 2.5), the vast majority of them did not explicitly consider usability. What is worse, there were cases in which some potentially negative usability impacts introduced by the countermeasures were clearly apparent such as where users need to use a wearable device like in [115, 71], and where the SPA capability might be restricted [56]. Future work should conduct rigorous and systematic studies of the usability of the countermeasures already proposed to assess how usable they are. Beyond these usability studies of existing countermeasures, future work on SPA security and privacy mechanisms should also consider usability from the onset, not as an afterthought. For instance, novel SPA security and privacy mechanisms should avoid requiring extensive user involvement. Otherwise, it has been shown they may not be used [250]. A potential avenue to explore as future work regarding this example could be the AI-based techniques discussed in Section 2.6.5, which could be leveraged to predict users' preferences and help users set security and privacy controls much easier and with less involvement.

2.6.9 Profiling Attacks and Defences

Regarding profiling, we can clearly see in Table 2.1 that few attacks have been reported on this. Some of these attacks make some hard assumptions, like having access to all cloud data about a user through their user account. We believe that further research is needed to assess whether other types of more sophisticated profiling could be conducted with access to less information. Furthermore, the community needs to understand whether tracking, which is pervasive across the web [150], could also apply and be feasible across the SPA ecosystem.

In terms of defences, we can also see in Table 2.2 the lack of work in this area. Some of the challenges we mentioned before would indeed help alleviate profiling, such as user awareness and usable controls (Sections 2.6.7 and 2.6.8), systematic privacy assessments (Section 2.6.6), and knowledge-based AI techniques to express/verify norms about how data are collected and use of data across the SPA ecosystem (Section 2.6.5). However, other open challenges would remain, and profiling-specific countermeasures are also needed. For instance, SPA traffic needs to be properly obfuscated and masked to encode users' interaction with the devices in addition to the existing encryption mechanisms already in place. Note that current encryption mechanisms are not sufficient to avoid traffic profiling as shown in [30].

Beyond differential-private approaches like the countermeasure introduced earlier [137], one possible avenue would be to adapt existing mechanisms to the case of SPA, such as traffic morphing techniques [242] to prevent statistical traffic analysis. This can be done by altering one category of traffic to look like another one. However, this and most other existing traffic analysis countermeasures are vulnerable as they only obfuscate exposed features of the traffic by muffling these features and adding dummy packets. Thus, they are unable to prevent the leakage of many identifying information [67]. Another avenue could be based on mix networks [227] and/or onion routing [160]. However, both of them may also be vulnerable to attack. For instance, mixing is susceptible to long term correlation and sleeper attacks [227], and onion routing is susceptible to an adversary correlating the traffic [226] and to misconfigured and malicious relays [106].

2.7 Focus of the PhD and Most Related Work

As remarked in Section 2.4, SPA skill is one of the key architectural elements offering a large attack surface. Hence, assessing and mitigating the vulnerabilities inherent within this element is crucial to have a secure and privacy-aware SPA. In this regard, the rest of this thesis focuses on the SPA skill ecosystem. In particular, the systematic measurement and privacy assessments of SPA skills as this is one of the key open challenges in this domain (c.f Section 2.6.6). Systematic measurement and privacy assessments of SPA skills is important to answer key questions such as: 1) what are the actual data collection practices of third-parties skills? 2) how transparent are these data practices? 3) do users have control over the amount and nature of data being collected? And 4) how effective are SPA market operators in helping protect users from malicious third-party developers? Furthermore, a systematic study of the SPA skills is vital to measure how many privacy violations are there in the wild of the third-party ecosystem and the extent of such violations. In addition, a longitudinal study could be essential in understanding the SPA skill ecosystem better and comprehending the type of skills available, their capabilities, how they are being used, and who is behind them (number of third-party developers, etc.). We next provide a brief review of the most related work to this study.

We distinguish two key areas of related work: 1) skill measurement and privacy assessment; and 2) traceability and privacy policy assessment.

2.7.1 Skill Measurement and Privacy Assessment

There has only been limited related work on SPA third-party skills and the mechanisms they use in the discovery process. The author in [239] studied the challenges related to third-party skill discovery process and recommends ways to make skills more accessible by using contextual and personal signals, among other means, rather than via trial and error invocation. There is also research on measuring the availability of privacy policies for skills. Most noticeable is the work of Alhadlaq et al. [5] that analyse the Alexa skill store where their finding shows that 75% of the skills do not have a privacy policy. However, the most important limitation lies in the fact that the study does not consider the personal information collected by the skills in their privacy assessment.

In another work conducted in parallel to this study, Liao et al. [136] performed systematic skill measurement and used skills description as a baseline to detect inconsistent privacy policies. The findings in this work are subject to at least three limitations. First, using skills description as a baseline to detect inconsistent privacy policies is ineffective as many developers don't often mention what personal information they collect when describing their skills. For example, skills developed by *Vipology* such as “100.7 BIG”, “FM101.7”, “KISS FM 101.7”, request for *Device Country and Postal Code* through Alexa API. However, these permissions are not mentioned in the skill descriptions. Secondly, the permissions mentioned by a developer in the skill descriptions could be different from what the skills are actually requesting. A good example is the “Interflora” by *Interflora British Unit.* This skill collects *Full Name*, *Email Address*, *Mobile Number*, and *Amazon Pay* permissions through Alexa API but only mention the Amazon pay permission in its skill description. Thirdly, the work does not consider the collection of personal information via conversations which is one way of bypassing APIs for data collection.

Large scale measurement study was also conducted by Guo et al. [88] to understand skills' behaviour. The authors in this work investigate 30K skills by building an interactive system called *SkillExplorer*. This system explores the interactive nature of skills and identifies those that request personal information through conversations bypassing developer specifications. Authors in [247] extend this work with *SkillDetective*, to 52 different policy requirements in a broader context from multiple sources including textual, image and audio files. While these studies identified several skills asking for personal information without conforming to the privacy requirement, the proposed systems only work with those skills that have a unique invocation name, so they missed thousands of skills with similar or the same invocation names.

There has also been a study in [44] on whether the vetting process provided by SPA providers can be compromised. Over a period of 15 months, the researchers crafted and submitted for certification 615 mock policy-violating skills comprising of 234 Amazon Alexa skills and 381 Google Assistant actions. Interesting, the authors got all the Alexa skills and 39% of Google actions certified. This study provides evidence that the current skill certification process in leading SPA platforms needs improvement to meet users' expectations of privacy. Authors in [208] also systematically assess the attack surface of SPA by looking into whether the SPA skills voice commands are intended to insert action or retrieve information and whether the command is sensitive (e.g. stop the camera, show me the last activity from the front door). Their result shows that while the percentage of sensitive skills is small, it gradually increases over time. However, the authors do not study whether such a command is intended to collect personal information from the user or check how well the skill discloses its data practices for such collected personal information.

In a more recent work [134] done after our study, the authors crawled skills across seven markets and similarly characterised them as we later do in Section 3. In addition, they studied the feasibility of conducting squatting attacks and provided an initial look at the effectiveness of skill privacy policies. The authors propose an automated method based on PoliCheck [24] to detect privacy inconsistency at scale. Their system achieved an accuracy of 83.3%, and their finding shows that 77% of the Alexa skills with data permissions adequately disclose their data practices in their privacy policies. However, the most important limitation of this study lies in the fact that PoliCheck was trained *only* with android privacy policies, and the ontology used for PoliCheck only considers a subset of the data types in the skill ecosystem.

All of the previous studies [5, 134, 88, 136] performed their measurement using just a single snapshot in time, which is not sufficient to understand the ecosystem. As early mentioned, a longitudinal study of skills across time is needed to adequately track the evolution of this ecosystem and propose better ways to improve it.

2.7.2 Traceability and Privacy Policy Assessment

Traceability analysis is useful to evaluate whether the transparency and privacy control mechanism provided to the users are congruent with documented privacy policies.A number of studies [155, 256, 23, 28, 27, 237] have investigated the traceability between the actions of the data specified in the privacy policies of third-party applications and the related data operations evident to users. For instance, the study described in [248] focused on ensuring software requirements compliance with governing legal texts and privacy policies. Authors in [256] analyse Android apps characteristics for likely non-compliance with privacy requirements and some selected and applicable laws. They implement a machine learning-based privacy policy with static code analysis of Android apps to detect potential non-compliance with privacy requirements. Their findings show that 71% of apps that lack a privacy policy should have one, and a substantial portion of apps exhibit potential privacy requirement inconsistencies. In a different study, researchers in [23] also propose a tool, PolicyLint, to identify policy contradictions in Android. The authors analyse the policies of 11,430 android apps and find that 14.2% of these policies contain logical contradicting statements that may mislead users and create privacy issues.

In another study, Wang et al. [237] created a hierarchical mapping-based approach for privacy policy analysis that can handle user-inputted data as well as data accessed directly through the mobile device. The user input data is checked for potential privacy leaks, and this information is used to determine whether the app's privacy policy contradicts the leakage. In addition, a data flow analysis is used to verify the consistency between the data collected by the app and the privacy policy provided.

Other works have likewise analysed privacy traceability in domains such as Online Social Networks [28, 27], and Social Media Aggregators [155]. The study in [27] posited a method to help developers automatically map traceability between policies and operationalised controls and maintain it through the development cycle. They proposed Castor, a tool capable of inferring semantic mappings with F1 accuracy of 78% for Friendica and 70% for Diaspora social networks. Authors in [155] also assessed the privacy of 13 popular Social Media Aggregators (SMAs) from 3 app stores. By inspecting the mobile and social media data accessed by the apps, checking for privacy policies and their compliance with distributors' vetting policies, and then performing a qualitative assessment of traceability between privacy policies and the actual transparency and control mechanisms offered to users by the apps' interfaces, the authors identify privacy violation and lack of transparency in 5 of the SMAs.

However, no previous study has focused on analysing the privacy traceability of SPA skills considering personal information collected via APIs and through conversations. Nonetheless, the above-related works have posited methodologies that could be adopted to measure privacy policy traceability in the SPA domain to understand better, the different risks the ecosystem presents and aid in formulating appropriate security and privacy policies for the users.

2.8 Conclusion

This chapter analyses and classifies the security and privacy issues associated with SPA and how a range of malicious actors can exploit them to harm the security and privacy of end-users. We showed that the attack surface of this increasingly popular technology is vast and that the interaction between the users and the SPA devices is currently the weakest link. However, we have identified a wide range of attacks that can put users at stake. In as much as there is no single panacea solution for all security issues, the proper understanding of security pitfalls will go a long way in enabling manufacturers, researchers, and developers to design and implement robust security control measures.

State of the art review like this could help researchers prioritise the most promising areas to improve our understanding of attacks on SPA and to devise usable ways to counter them. Although there is already very active research on securing intelligent assistants, few of the approaches consider the whole picture of the complex architecture SPA have. We particularly highlighted open challenges that we deem of critical importance, including making authentication stronger, enhancing authorisation models and mechanisms, building secure and privacy-aware speech recognition, conducting systematic security and privacy assessments, developing AI-based security and privacy countermeasures, improving user awareness and usability, and studying further profiling attacks and defences.

We have looked at the relevant literature on SPA security and privacy issues and identified open research direction, fulfilling the first objective. The next chapter presents a systematic measurement of third-party skills of SPA, which has been identified as one of the architectural elements offering a large attack surface and needing more attention. This is crucial for assessing and mitigating vulnerabilities inherent within this element.

Chapter 3

Systematic Study of Third-Party Skills

This chapter focuses on the work done to achieve OBJ2 presented in Section 1.2: *To develop a practical method for detecting third-party voice applications with concerning privacy practices.* In view of this, this chapter presents a systematic measurement of SPA skills, which has been identified as one of the architectural elements offering a large attack surface and needing more attention. We perform a measurement of the Amazon Alexa skill ecosystem — the largest SPA in terms of the number of skills by far and analyse skills in the look for concerning practice. We studied developers' practices including how they collect and justify the need for sensitive information, by designing a methodology to identify over-privileged skills with broken privacy policies. We uncovered bad privacy practices in about 43% of the Alexa skills that use permissions involving 50% of the developers with skills that request permissions. The findings led to a responsible disclosure process where we reported 675 Alexa skills with privacy issues to Amazon and the affected developers.

3.1 Introduction

The skills ecosystem widens the attack surface of SPA, as malicious third-party actors may develop potentially harmful or unwanted software [212, 127]. In addition, the enormous amount of personal data skills could collect opened up the SPA to another avenue of attack. Unfortunately, it is unclear what are the risks third-party skills may pose to users, beyond the potential for skill impersonation [252, 127] and other potentially suspicious behaviours [88]. Authors in [44] have very recently shown that the

Amazon skill certification process can be subverted by crafting mock policy-violating skills. Hence, there is a critical knowledge gap in understanding what are the *actual* data collection practices of skill developers in the wild. This gap prompts three core research questions:

- RQ3.1 — What is the current state of affairs in the third-party ecosystem of skills?
- RQ3.2 — Is the collection of personal information explained?
- RQ3.3 — Can we pinpoint fine-grained privacy issues (i.e., at the permission level)?

In this chapter, we perform a systematic measurement study of Amazon Alexa skills across the entire marketplace of 11 different countries to shed light on the third-party skills ecosystem and developers' practices. We characterise skills using their market categories, names, and developers, among others. We also characterise the data permissions they request during runtime and their traceability with data practice statements in the skill's privacy policy. We then analyse suspicious skills in the look for concerning privacy practices. We do this by systematically analysing the traceability between the permissions requested by the skill and the data practices stated by the skill developer in the privacy policy. This is done following a black-box approach since the skills' code, or executable files are not available — skills *run in the cloud instead of the users' device*, e.g., as an AWS Lambda function [18] or in a server controlled by the skill developer [19].

3.1.1 Threat Model

The attack surface of SPA is considerable and can lead to several security and privacy issues. One of the attack entry points in SPA, and the focus of this chapter, is SPA skills. Unlike mobile apps, skills do not run on any user-controlled device. Instead, skills either run in the Amazon cloud (e.g., as an AWS Lambda function) or in a server controlled by the developer [19]. This could allow a malicious developer to sneak malicious code into their software via the application backend [212]. They could manipulate skills to covertly introduce misperception about reported events [206] and also use skills with duplicate invocation names, similar phonemes, or paraphrased invocation names to impersonate another skill [252, 127]. Recently, there has been a report of how malicious actors, through a phishing attack, could steal a token that can let them add suspicious skills to the users' account and access voice history [36].

In this study, we also consider the threat from skills developed by third-party developers but focus on the data practices of such developers, whether malicious or just negligent. Skills allow users to interact with their services and require data exchange. This could be data that are part of the user's account with Amazon, such as users *location, mobile number, email address, name, address* [14], sensitive data inferred from user actions with the skills, or personal data collected by the skills during their conversations with the users. By considering skill developers as adversaries, we look into potentially over-privileged skills to understand what personal information they collect. This issue has been explored by several previous studies on permission gaps in other domains [35, 235] but not in SPA and their skills. While declaring more permissions than required seems to have no impact on a skill's functionality, it could, however, be leveraged by a malicious developer to achieve malicious goals [35], which could impact the users' privacy.

This chapter presents a method that could detect skills with inappropriate usage of data permissions and help protect users' privacy. For instance, "Mock Interview" skill *by Graylogic Technologies* as detailed later. Unlike other prior study [134, 127, 252], our work offers a much more nuanced view of the data practices in third-party skills and their developers. This allows us to uncover many more partial traceability cases. Our crawling strategy also allows us to analyse the traceability of more unique skills.

3.2 Data Collection

Unlike other platforms like Android, Amazon Alexa runs the skills in the cloud and *code of the skills is not publicly available.* To analyse the skills ecosystem, we built a Web scrapper with a framework that recursively crawls all markets with third-party skills and extracts metadata published by Amazon about the skills. Amazon operates 14 separate online marketplaces that cater to a variety of segments. Out of which 11 have an online store with third-party skills. We crawl these 11 markets: The United States (US), United Kingdom (UK), India (IN), Australia (AU), Canada (CA), Germany (DE), Japan (JP), Italy (IT), Spain (ES), France (FR), and Mexico (MX). Most importantly, skills are only available to registered users, and a user can only be linked with one marketplace at a time.

The Web scrapper visits the different skill categories while building a collection of links corresponding to each skill. It then iterates through the collected skill links to visit the skill's website and extract the skill attributes. However, we see a lack of coverage when deploying traditional crawling methodologies. Amazon lists the top most popular skills organised by category (and subcategory) up until a maximum of 400 pages per category (23 categories and 66 subcategories in total). However, the skill marketplace is not entirely listed in the Amazon Alexa index. Thus, crawling this index alone does not make the data collection exhaustive.

We overcome this limitation by looking into subcategories for each category per marketplace. Every subcategory across marketplaces returns less than 400 pages so that we can crawl all the skills within them. There are only two subcategories with more than 400 pages, both in the US (out of a total of 66 subcategories): Knowledge & Trivia (from the Games & Trivia category), and Education & Reference (from the same category). For these two specific cases, we then use a best-effort approach and re-crawl skills in those subcategories with a different sorting of the skill listing (i.e., ordering by *average customer review* in addition to the default one — *featured*). Thus, we are confident that we capture most, if not all, of the market space, as we collect all the skills available across all 11 marketplaces but the US. In the US, we collect more skills than the sum of all approximate numbers Amazon gives per subcategory,[1] which is ≈57,000 at crawling time (see below for the date) , and we are able to crawl 61,362 skills from the US market, as detailed later.

Additionally, due to a vested interest in protecting its data, Amazon discourages scraping [13] and currently implements different anti-scraping techniques, making it challenging to scrape Alexa skill markets. These include the use of captchas, email verification, and blacklisting of IPs.[2] Besides, some of the Alexa markets have varying page structures. Amazon instead encourages using their APIs such as the Product Advertising API and Marketplace Web Service Products API to query the marketplace. However, Amazon is selective regarding the information that can be accessed from these APIs. To overcome some of the anti-scraping measures:

1. We limit the rate at which we generate our requests;
2. We mimic human behaviour;
3. We make our requests through a pool of IP addresses and proxies;
4. Lastly, we design our scraper to handle and react to exceptions such as "Element-NotVisibleException", which occurs when the scrapper tries to find an element not visible within the skill page, or "NoSuchElementException" when elements unexpectedly become not available.

[1] Note that Amazon does not provide the exact number of skills per category/subcategory when they contain 1,000 skills or more, but it rounds it down to the closest number in units of thousands.

[2] Recent judgement in the US shows that such scraping from public services is legal [202, 103].

While new skills are added daily, we base our study on those skills accessible to users between the 15th of July 2020 and the 29th of July 2020. This is the time frame that took our crawler to visit all marketplaces. Since our data collection requires issuing a request to web services, we avoid generating unnecessary traffic that may affect their normal operations and limit the rate at which we generate our requests.

3.3 Characterisation

In this section, we present our characterisation of all Alexa skills across 11 markets.

For all the skills we collect, we characterise the market category they belong to, the utterances that activate the skill, and the developer that has created it. Two additional key elements we extract are i) the permissions that the skill requires to access personal information from the Amazon user's account **via the Alexa API at runtime**, which Amazon makes publicly available in the marketplace; and ii) the privacy policy declared by the developers. We then use these two elements, together with an interactive analysis, to study when there is a privacy violation or a concerning practice that may harm users. In summary, we use the following attributes (which we scrape from the marketplace) to characterise every skill: invocation name, permissions, the category, developer's information, privacy policy, terms of use, skill description, rating information and reviews.

Note that one skill can belong to different categories and be hosted in multiple marketplaces. In these cases, the URL may have different parameters, but the path displays a unique identifier per skill. We identify unique skills posted across markets in a process we call de-duplication.

3.3.1 Skills and Developers

Table 3.1 shows the breakdown of the total number of Alexa skills and developers across Amazon marketplaces. Overall, we see 199,295 skills published by 88,391 developers. Note that most of these skills have been developed in just a few years.[3] English-speaking marketplaces host the highest number of skills and developers. On the other end, the smallest market is Mexico with 1,972 skills and representing 1% of our dataset. However, there is overlap across markets as developers can publish the same skills in different markets [20]. This overlap is larger among English-speaking markets such as the US, UK, CA, and AU, where Amazon tends to migrate existing skills [10]. In particular, we

[3] Alexa had only 135 skills in early 2016 [180].

observe that close to 20,000 different skills in the UK market are also in the US market. When we de-duplicate skills, we observe that 53.76% (47,520) of the developers and 56.2% (112,029) of the skills we collect are unique. In what follows, we look at unique skills unless we analyse specific markets.

Table 3.1 Number of skills & developers. English-speaking markets represent 82.74% (92k skills and 36k developers)

Marketplace	Skills		Developers	
	N	Percent	N	Percent
US	61,362	30.79%	27,030	30.58%
UK	32,822	16.47%	14,487	16.39%
India	29,344	14.72%	12,823	14.51%
Canada	26,027	13.06%	11,509	13.02%
Australia	23,909	12.00%	10,854	12.28%
Germany	9,096	4.56%	3,385	3.83%
Spain	4,759	2.39%	2,441	2.76%
Italy	4,203	2.11%	2,049	2.37%
Japan	3,513	1.76%	1,407	1.59%
France	2,288	1.15%	1,194	1.35%
Mexico	1,972	0.99%	1,212	1.37%
Total	199,295	100.00%	88,391	100.00%
Unique	112,029	*56.2%*	47,520	*53.76%*

We illustrate the relationship between developers and skills per marketplace in Figure 3.1. While most developers publish only one skill, a few have tens, and a handful has hundreds and even thousands of skills. In particular, 125 developers have published more than 50 skills, 69 developers more than 100 skills, and 5 developers more than 1K skills. We then study the description of all the skills published by prominent developers and see that many implicitly indicate that the skill has been developed with an automated platform. This is usually done by referring to the name or the URL of the framework that produced the skill. Popular among them are *VoiceApps.com*, *VoiceXP.com* and *VoiceFlow.com*. These platforms provide free sample skills that developers can customise regardless of their technical background. In particular, they offer visual, drag-and-drop editors that lowers down any development barriers. We mine all descriptions in our dataset and see that at least 7% of the skills have been developed with these platforms. In particular, we observe a ratio of 1 developer to 30 skills with the most popular automated platform (VoiceApps.com).

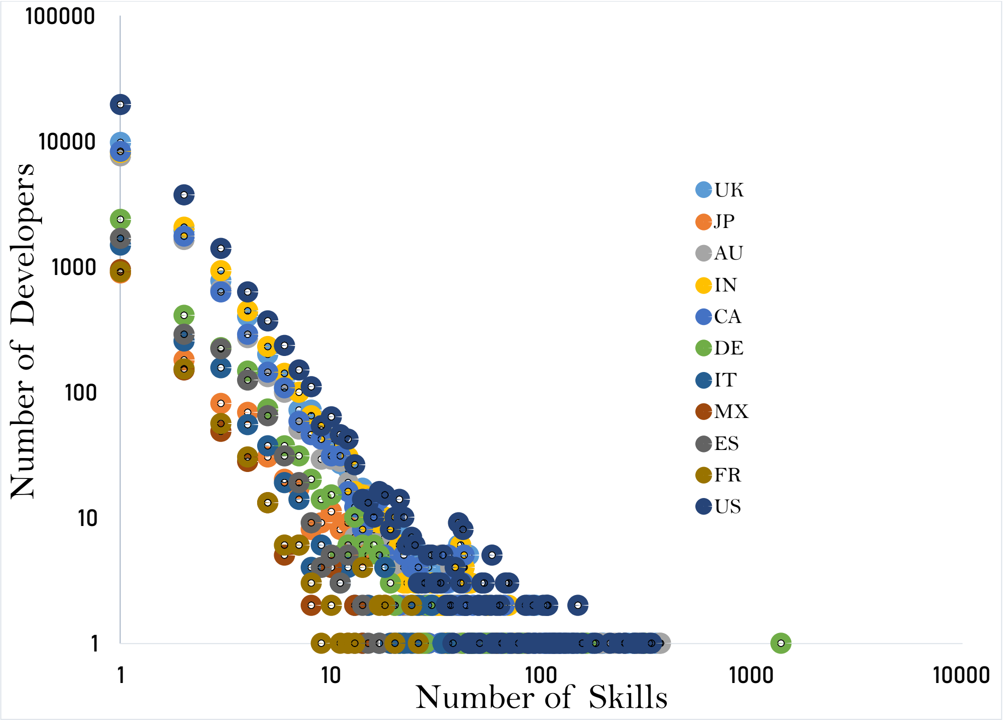


Fig. 3.1 Developers vs the number of skills they publish per marketplace

3.3.2 Skill Invocation Names

Previous research showed that an attacker could impersonate skills by crafting specific malicious skill invocation names [252, 127, 44] by reusing the same as or (phonetically) similar invocation names to other skills. We measure the prevalence of same and similar skill invocation names across the Alexa ecosystem to understand the potential for this risk. We find 24,468 unique skills (21.8%), 6,876 (11.2%) in the US, with the same invocation name as another/other skill/s after cross-market de-duplication. We also find a considerable amount of very similar skill invocation names. For instance, we see that between 5% and 15% of all the skills in the US have a different but phonetically similar name. This is in line with recent measurements in less markets [134].

3.3.2.1 Invocation Name Reuse

There are 99,521 skill invocation names out of all 112,029 unique skills in our dataset. Figure 3.2 shows a scatter plot with the aggregate of invocation names per marketplace. In particular, the figure shows the total number of skill invocation names (Y axis) w.r.t. how many other skills have the same invocation name (X axis). We see that the number of skills with unique invocation names ($X = 0$) ranges from 1,914 (Mexico) to 54,486 (US) and totals 66,087 after cross-market skill deduplication. Recall that we only consider unique skills when aggregating results across marketplaces. When we look at $X > 0$, we see 24,468 skills with invocation name reuse (10,480 developers reuse 24,468 skill invocation names). As values of X increase, we see more popular names. An important number of skills share the same invocation name in different markets. We observe that India predominantly appears in cases with large values of X, with about 34% of their invocation names being reused.

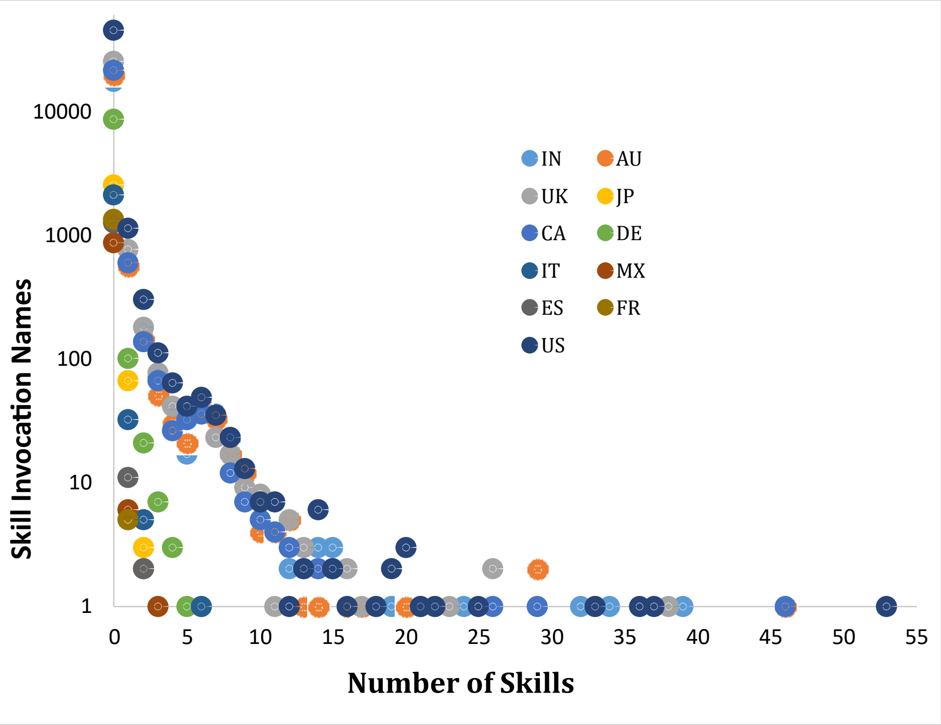


Fig. 3.2 Number of skills with the same name across Alexa marketplaces. Most of the skills use a unique invocation name, but a high proportion of skills use the same invocation name

Next, we study English-Speaking Markets (ESM) alone. Table 3.2 shows the number of skills that reuse names by market category in ESM. Games and Trivia turn out to be the category with the highest number of reused skill invocation names. We also observe

968 skills with the same skill invocation name and developer name in a single market (IN-280, CA-121, DE-36, UK-192, ES-14, FR-8, US-279, AU-38). One good example is the "Africa Facts" in the UK market by *Yasemin Woodward* with skill ids B07PGN7T9D, B07PHRMML4, B07PK4GNYC, B07PKVMYTB, B07PR67XTQ, and B07PFQ39Y6. This suggests some developers might be attempting a form of *sybil attack* [65], where they try to have multiple skills with the same invocation name (but different ID) to increase the chance for one of them to be selected.

Table 3.2 Number of reused skill invocation names per category in the English-speaking markets

Category	Invocation names	%	Skills	%
Games & Trivia	7,251	30%	26,580	29%
Education & Reference	4,250	17%	13,863	15%
Music & Audio	3,814	16%	12,121	13%
Lifestyle	1,697	7%	6,074	7%
Smart Home	1,130	5%	3,456	4%
Health & Fitness	977	4%	3,505	4%
Food & Drink	597	2%	2,111	2%
Business & Finance	783	3%	2,494	3%
Kids	683	3%	2,356	3%
Others	3,286	13%	15,235	22%
Total	24,468	100%	92,451	100%

Figure 3.3 shows the most popular skill invocation names in the 5 English-speaking marketplaces. "Cat Facts" is the most popular skill invocation name used 171 times. Likewise, the word "Fact" is used by 1,979 developers appearing 3,905 times across all marketplaces. We observe that many fact skills are single interaction skills (they terminate their interaction after performing one task), out of which over 40% are developed using automated platforms. This may be related to these platforms offering free fact skill templates, as discussed above.

3.3.2.2 Invocation Name Similarity

Because of the slight differences in invocation names, we also sought to measure how similar skill invocation names are. To do this, we use the Levenshtein edit distance [135]

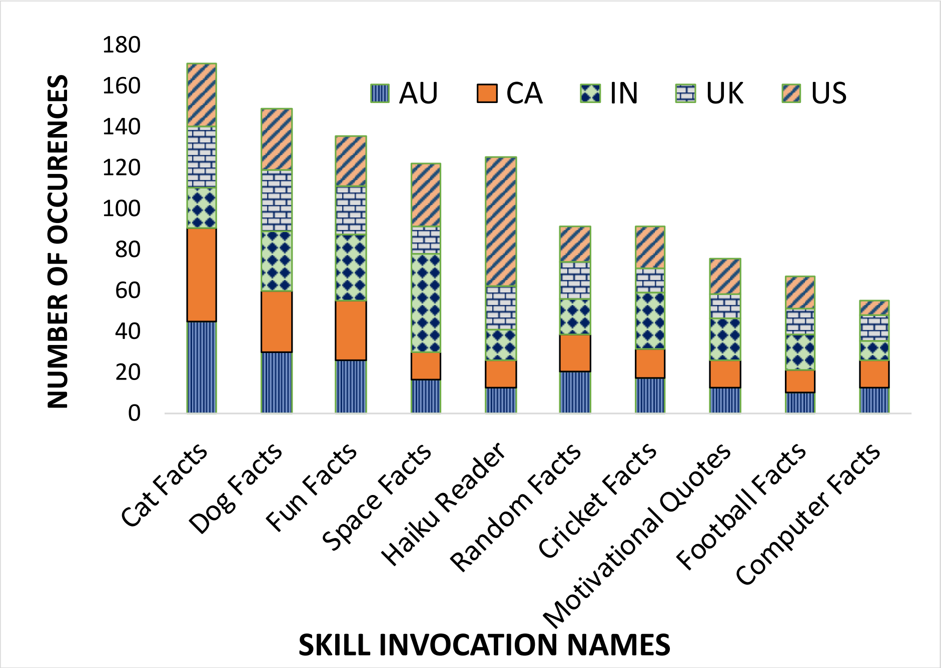


Fig. 3.3 Top 10 skills invocation names (English-speaking)

to compute the similarity between skill invocation names. We also consider the phonetic similarity between invocation names [233], i.e., the phonetic transcription of an invocation name. This is because both textual and phonetic invocation name similarity may be leveraged to exploit the speech recognition capabilities of SPA for an impersonation attack, known as skill squatting [252, 127].

We use the open-source Carnegie Mellon University Pronouncing Dictionary (CMU-dict) to get the phonetic transcription of words.[4] For every skill in every English-speaking market, we lowercase its invocation name, remove all non-ASCII characters and punctuation, and replace digits with their equivalent texts (e.g., "1" becomes "one"). Afterwards, we use the CMU dictionary to obtain the phonetic transcription of every skill invocation name, ignoring those skill invocation names without phonetic transcription. For every resulting skill, we estimate the shortest phonetic Levenshtein distance to any other skill invocation name in the same market to identify skills with similar-sounding invocation names. In the process, we factor in the dynamic lengths of different skills invocation names by normalising the Levenshtein distance by the length (in phonemes) of the longest skill invocation name compared.

Table 3.3 shows the quantity and percentage over the total number of skill invocation names we found with a phonetic translation and a minimum edit distance when compared

[4] https://github.com/cmusphinx/cmudict (version 0.7b).

Table 3.3 Skill invocation names with Levenshtein distance ≤ 0.2 across English-speaking markets

Market	$lev \leq 0.1$	$lev \leq 0.2$	Total
US	3,830 (8.82%)	11,929 (27.47%)	43,410
CA	1,514 (7.88%)	4,836 (25.17%)	19,209
AU	673 (6.86%)	4,439 (24.69%)	17,974
IN	3,048 (14.25%)	6,699 (31.31%)	21,389
UK	1,714 (7.80%)	5,321 (24.24%)	21,950

to the other skills of ≤ 0.1 and ≤ 0.2 across the five English-speaking marketplaces. We observe that in all English-speaking marketplaces (except India), we have an average of 6 to 8% skills invocation names with minimum Levenshtein distance ≤ 0.1, and around 26% with minimum distance ≤ 0.2. Moreover, most of the skills with lower phonetic Levenshtein distances ≤ 0.1 are plural versions of the original skill, such as ”Fun Fact Quiz” and “Fun Facts Quiz” (India), or “Panda Facts” and “Panda Fact” (US).

However, there are others, such as “Sweet Facts” and “Wheat Facts” (US), which would have a higher edit distance without transcription but that when compared after transcription have a Levenshtein distance ≤ 0.1. In either case, we can see a considerable amount of skill invocation names that are quite similar to each other across the five English-speaking markets. For instance, we see that in the US, 5% of skills (3,215) have a Levenshtein edit distance (after phonetic transcription) ≤ 0.1, and 15% of skills (7,332) have a distance ≤ 0.2%.

3.3.3 Characterisation Take-aways

Take-aways. Our market characterisation provides a high-level overview of the skill ecosystem and how Alexa marketplaces are structured. In summary, the main takeaways include:

1. The skill ecosystem has grown considerably fast in recent years, and it currently has few developers with thousand of skills each. English-speaking markets generally dominate and push skills to other markets.

2. We see skills developed with automated platforms. The use of these platforms lower down the development barriers and enable bulk skill creation.

3. We observe a high prevalence of skills with the same or similar name, with the associated potential for impersonation and Sybil attacks as shown in previous works.

3.4 Permissions and Traceability

In this section, we first explore the data permissions in all Alexa marketplaces, analysing which permissions are more often requested by Alexa skills and developers among the different markets and how they are distributed between skills. Then, we look at skill privacy policies to understand how developers disclose and justify the data permissions they ask for. To do so, we create the **largest traceability dataset for Alexa** known to date, namely the traceability-by-policy dataset (TBPD), which includes the traceability analysis of 1,758 skills requesting data permissions and their policies from the five English-speaking marketplaces (Australia, UK, US, Canada, and India). Using the dataset, we analyse the skill traceability at the skill category and developer dimensions and identify developers' practices.

3.4.1 Data Permissions

Skills can request access to user information through the Alexa skill API. This information becomes available on a per-skill and per-data-type basis when the user consents. To manage the consent: i) Alexa declares the permissions of skill on their store and gives users control through the Web or the mobile app, and ii) skills have to inform users in their privacy policies how they will use the personal information they collect. Alexa currently supports 11 types of permissions [14]. These are: *Device Address* (15.0%), *Location Services* (4.1%), *Email Address* (15.7%), *Device Country and Postal Code* (13.7%), *Reminder* (9.1%), *Customer Name* (10.3%), *List Read Access* (5.3%), *List write Access* (4.9%), *Mobile Number* (4.3%), *Amazon Pay* (1.8%), and *Skill Personalization* (0.0%).

At the time of our analysis, we do not see any skills with *Skill Personalisation* yet. Also, our analysis reveals *Notification* (15.5%), which is now deprecated and replaced by *Reminder* and *Timers* (0.3%). The most prevalent permissions are generally used to offer services based on the user's location. Figure 3.4 lists the 13 Alexa permissions together with the number of skills that use them per market.

Overall, 2,608 (2.3%) skills across markets ask for data access permissions. This could, in principle, be seen as good news. Still, 2.6K skills is a sizeable set of third-party

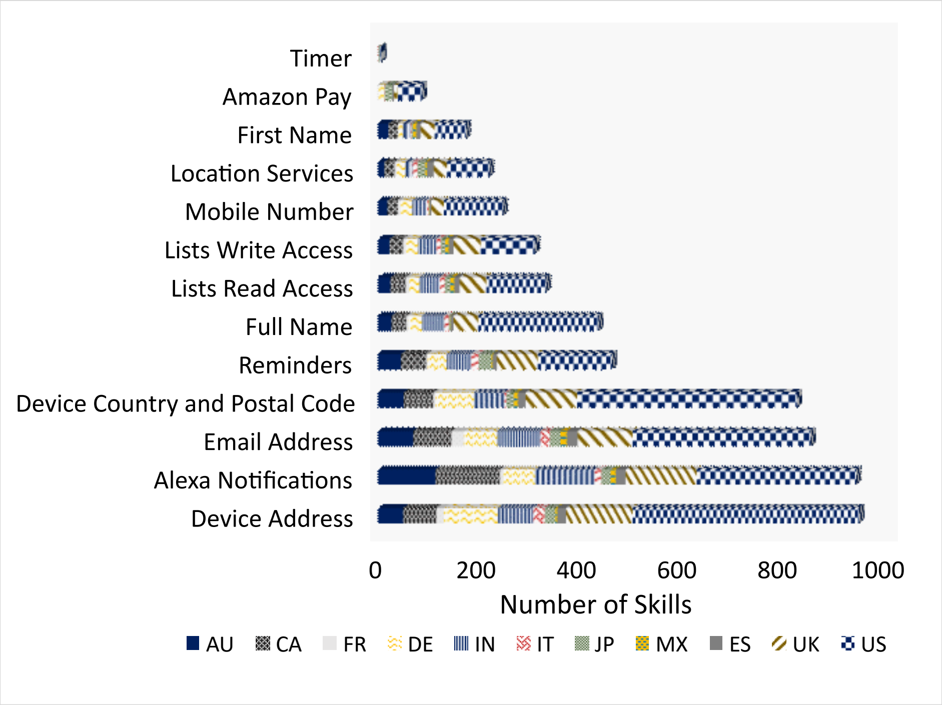


Fig. 3.4 Permissions distribution across the markets

software accessing personal information, and the practices of their developers require scrutiny. We also argue that skills may collect personal information using other means, i.e., through account linking or conversations, as discussed in Section 3.5.

Table 3.4 shows the permission distribution of skills per marketplace. As shown, most skills ask for one permission (typically, the device address as discussed above), with all marketplaces showing a similar pattern overall — albeit with slight differences between those markets with more or less number of skills. There is also an important minority of skills asking for two or three permissions. Finally, there are a few exceptions where skills ask for over three permissions. The most noticeable one is a skill called “BILH Staff Chatbot” by *BIDMC* (the Beth Leahy hospital). The skill is available in the US, and it is requesting several unique permissions (i.e., *Name*, *Email Address*, *Mobile Number*, *Reminders*, *Location Services*, and *Device Country and Postal Code*). The skill is a chatbot for employees to help them fill out a COVID-19 staff daily health questionnaire. While the skill is not intended to diagnose users (according to the description), it asks for 13 different symptoms. The use of this skill poses an important privacy risk to employees, because the Web form version of the questionnaire does not ask for data such as postal code or location, so it is not clear why these are needed in Amazon.

Table 3.4 Permissions distribution by skills and marketplace

		No of Unique Permissions									
		1		2		3		4		≥5	
Markets	Skills	N	%	N	%	N	%	N	%	N	%
US	1,698	1,169	68.85%	301	17.73%	143	8.42%	51	3.00%	34	2.00%
UK	581	418	71.94%	120	20.65%	26	4.48%	7	1.20%	10	1.72%
CA	366	262	71.58%	74	20.22%	14	3.83%	6	1.64%	10	2.73%
IN	358	250	69.83%	66	18.44%	20	5.59%	7	1.96%	15	4.19%
AU	348	245	70.4%	78	22.41%	10	2.87%	5	1.44%	10	2.87%
DE	341	236	69.21%	83	24.34%	12	3.52%	7	2.05%	3	0.88%
JP	126	100	79.37%	21	16.67%	4	3.17%	1	0.79%	-	-
ES	89	56	62.92%	24	26.97%	6	6.74%	3	3.37%	-	-
IT	85	58	68.24%	21	24.71%	5	5.88%	1	1.18%	-	-
FR	63	44	69.84%	15	23.81%	4	6.35%	-	-	-	-
MX	46	30	65.22%	13	28.26%	3	6.52%	-	-	-	-
Total	4,101	2,868	69.93%	816	19.9%	247	6.02%	88	2.15%	82	2.00%
Unique	2,608	1,807	69.29%	499	19.13%	191	7.32%	67	2.57%	44	1.69%

3.4.2 Traceability Analysis

We look at privacy policies to understand how developers disclose the permissions they request. Specifically, we study the traceability between data operations obvious to users (recall users need to enable skills requesting permissions) and the data actions defined by developers in the skill policies. Note that Amazon's privacy requirements for skill developers mandate that a skill must come with an adequate privacy policy if it collects personal information. In particular, any collection and use of personal information need to comply with what is stated in the privacy policy [15].

To evaluate traceability between permissions and policies, we collect and tag what is the largest policy traceability dataset for Alexa known to date, containing the traceability analysis of more than 1,758 skills requesting data permissions in the five English-speaking marketplaces (Australia, UK, US, Canada, and India). We develop a Selenium module in Python that automatically visits and downloads every skill's privacy policy page, cleaning the HTML code to remove unnecessary markup code, normalising punctuation, and extra white spaces discarding non-ASCII characters and finally converting text to lowercase (this is later referred as the pre-processing phase). Afterwards, each skill is analyzed and evaluated as having *broken*, *partial* or *complete* traceability, following previous studies of traceability analysis in other domains [248, 28, 155, 256].

The type of traceability is identified by comparing the permissions requested by the skill through the Amazon Alexa API with the data practices covered in the skill policy. The different traceability types are explained next:

3.4.2.1 Complete

A skill is said to offer complete traceability if it provides adequate information in its privacy policy document about its data practices, i.e., the data action defined in the privacy policy document can be completely mapped to the access of data permissions. For instance, a skill like the "Aircraft Radar" in the UK market developed by *Chris Dzombak* offers complete traceability since it provides adequate information about its data practices in its privacy policy. The statement "aircraft radar uses your device's address to find your location and search for aircraft around you" can be mapped with the *Device Address* permission the skill collects.

3.4.2.2 Partial

This is when the transparency of data action defined in the privacy policy document maps partially to the access permissions. For example, when the privacy document states that a skill collects personally identifiable information without explicitly stating what this information is. A statement such as "we may require you to provide us with certain personally identifiable information" offers partial traceability since it does not explicitly state what information is collected. A skill is also said to offer partial traceability if not all its data permissions are covered in its privacy document. Partial disclosure of data practices may also occur when data practices in a skill privacy document are not well mapped with the skill's data permission. For instance, the statement "we may also collect your zip code" is partially traceable as the skills collect *Device Address*, and this refers to the user's full address, including the zip code and the street number. Examples that provide partial information in their privacy policy include: 1) "Vote Sam Feldt" by *Fanspoke* that collects *Mobile Number* 2) "Flight Level - An Aircraft Radar" by *O. Schafer* that collects *Device Country and Postal Code.*

3.4.2.3 Broken

If a skill requests permissions but does not have a privacy policy or the link to the privacy policy given is not working, we consider the traceability broken. Even when providing a privacy policy, a skill has broken traceability if it provides no data implication in its privacy policy document. For instance, when a skill collects the *Device Address*, and its privacy document states nothing related to *Device Address*, we mark such skill data practices disclosure as broken. We only mark a broken policy when permissions are not traceable to the data practices defined in the privacy document. In the UK marketplace, skills like "Daily Yoga" by *Siva Pandeti* collects the *Device Country and Postal Code*, "Bike Weather Man" by *Marc Easen* collect *Device Country & Postal Code*, and "Smoggy Alerts" by *Teal Dreams Software, Inc.* collects the *Device Country and Postal Code*, and they all exhibit broken traceability.

When analysing traceability, a skill requesting *Device Country and Postal Code* permission is tagged to have adequately disclosed its data practices if *Device Address* is mentioned in the privacy policy. This is because *Device Address* refers to the user's full address, including the *Device Country and Postal Code* and the street number. Also, we did not find any skills using *Skill Personalisation*, and grouped *List Write Access* and *List Read Access* under the *Personal Information* category as acknowledging personal information collection should be sufficient to disclose these permissions. Finally, *reminders/notifications* are not explicitly considered in the privacy requirements for skills by Amazon, as they may not have a personally identifying implication. Therefore, the traceability of all skills was evaluated considering: *Location Services*, *Amazon Pay*, *Mobile Number*, *Email Address*, *Name*, *Device Address*, *Device Country and Postal Code*, *Personal Information.*

3.4.3 Traceability Results

A total of 1,758 skills request data permissions in English-speaking marketplaces. Out of the 1,758 skills: 442 (25%) have broken traceability, 306 (18%) have partial traceability, and 1,010 (57%) have complete traceability. When we exclude the complete ones, we see 43% of the skills displaying bad privacy practices. Figure 3.5 shows the results by market, where we can see a very similar trend across marketplaces, with the UK appearing to have slightly more broken and less complete traceability results, but it also has the least number of skills and developers.

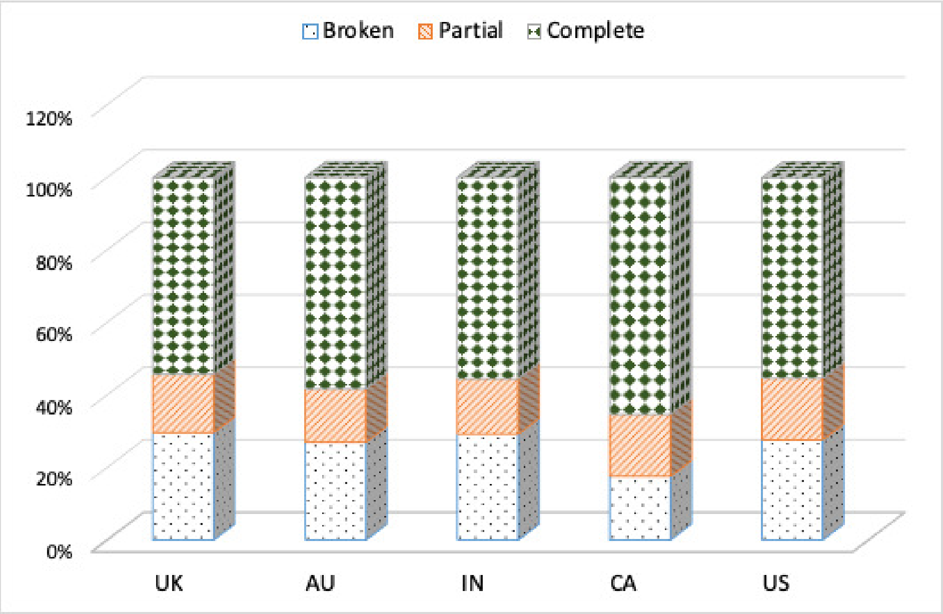


Fig. 3.5 Traceability for skills requesting data permissions in English-speaking markets

3.4.3.1 Traceability by Categories

To understand how traceability affects different types of skills, we first compare our results considering the 1,758 unique skills collected across different skill subcategories and English-speaking marketplaces, as shown in Table 3.5. Using subcategories, we show the breakdown to give a more fine-grained perspective of particular types of (or sector-specific) skills and associated traceability. We rank the different subcategories based on the number of issues (broken and partial) normalised by the number of well-defined policies (complete). Note that we do not list subcategories with less than 3 skills for the sake of clarity.

Sports has the largest proportion of skills with broken traceability, totalling 91% of all the skills in the subcategory. For instance, the “Western States” skill by *de Peck, Inc* collects the *Device Address* without any privacy document to disclose the data practices. Like many other subcategories, there are no skills with complete traceability in Novelty & Humour. When looking at the combination of broken and partially broken traceability, Games & Trivia is also one of the most problematic subcategories, particularly considering the number of skills it has. For instance, we find “Pixated Salat”, a prayer skill “to find Salah or Prayer time”. The skill is developed by *Pixated ltd*, a strategic advertising group, and asks for the device’s location. However, the link to the policy provided is broken, and it links to the main site of the advertising group instead (http://pixated.agency).

Table 3.5 Traceability per subcategory in English-speaking markets

Categories	R	B		P		C	
		N	%	N	%	N	%
Sports	1	43	91.49%	1	2.13%	3	6.38%
Social	2	98	85.96%	7	6.14%	9	7.89%
Novelty Humour	4	5	71.43%	2	28.57%	0	0.00%
Kids	5	10	83.33%	1	8.33%	1	8.33%
Games Trivia	6	89	57.42%	34	21.94%	32	20.65%
Schools	7	3	100.00%	0	0.00%	0	0.00%
Utilities	8	17	56.67%	5	16.67%	8	26.67%
News	9	30	63.83%	4	8.51%	13	27.66%
Health Fitness	10	60	48.78%	25	20.33%	38	30.89%
Organisers Assistants	11	7	63.64%	1	9.09%	3	27.27%
Lifestyle	12	74	46.54%	32	20.13%	53	33.33%
Weather	13	22	41.51%	13	24.53%	18	33.96%
Food Drink	14	39	42.86%	20	21.98%	32	35.16%
Streaming Services	15	4	57.14%	1	14.29%	2	28.57%
Productivity	16	42	51.85%	9	11.11%	30	37.04%
Business Finance	17	82	52.23%	16	10.19%	59	37.58%
Smart Home	18	44	57.14%	4	5.19%	29	37.66%
Travel Transportation	19	28	37.33%	18	24.00%	29	38.67%
Education Reference	20	59	43.70%	16	11.85%	60	44.44%
Movies TV	21	5	50.00%	1	10.00%	4	40.00%
Navigation Trip	22	2	40.00%	1	20.00%	2	40.00%
Connected Car	23	9	39.13%	3	13.04%	11	47.83%
Public Transportation	24	2	16.67%	4	33.33%	6	50.00%
Novelty Humor	25	3	25.00%	3	25.00%	6	50.00%
Home Services	26	5	19.23%	6	23.08%	15	57.69%
Self Improvement	27	2	50.00%	0	0.00%	2	50.00%
Wine Beverages	28	1	20.00%	1	20.00%	3	60.00%
Music Audio	29	94	29.38%	5	1.56%	221	69.06%
Shopping	30	26	20.63%	10	7.94%	90	71.43%
Calendars Reminders	31	1	14.29%	1	14.29%	5	71.43%
Pets Animals	32	1	33.33%	0	0.00%	2	66.67%
Event Finders	33	1	25.00%	0	0.00%	3	75.00%
Local	34	0	0.00%	1	25.00%	3	75.00%
Knowledge Trivia	35	1	8.33%	0	0.00%	11	91.67%

B = Broken, **P** = Partial, **C** = Complete, **R** (Rank) $\sim$ (B+P)/(C+1)

It is also important to note that even in categories that do not rank high, there are also skills with concerning data collection practices, like the Education & Reference category. This subcategory has a high proportion of complete traceability (44%), but it also has a very sizeable proportion of broken traceability skills (43%). For instance, in this category, we find a very interesting case of "A Sales Guy" by *VoiceXP*. This skill is supposed to provide information about Keenan, that according to his website, is a person who has been "selling something to someone for his entire life". Users that want to know about Keenan would have to give up their *mobile number, email address, full name, or device address.* The developer, *VoiceXP*, is a company that does so-called *voice domain registration services.* We refer the reader to Section 3.6 for further discussions on the type of concerning practices used by skill developers.

Regarding complete traceability, it is worth highlighting the bottom part of Table 3.5. Particularly, the *Music & Audio* subcategory has the largest number of complete traceability skills (221), which is also a sizable proportion of skills within the subcategory (69%). This is closely followed by Shopping, with many complete traceability skills (90) that make up 71% of skills within that subcategory. Note that these two categories relate to industries with a larger tradition of offering services on the Web, where privacy has been under scrutiny for longer. Still, adding up the skills with broken and partial traceability across the two subcategories gives a total of 135 skills. Therefore, even the best ranking categories have a sizable number of broken/partial traceability skills.

Finally, and rather interestingly, the Schools and Kids subcategories are ranked in positions #7 and #5, and have 100% and 0%, and 83% and 8% of their skills broken or partially broken, respectively, for a total of 11 skills. This is especially concerning as they may collect children's personal information, as we discuss more thoroughly in Section 3.6.

3.4.3.2 Traceability by Permissions

To understand how traceability varies across the different types of permissions, we also look at the traceability of skills per permission requested. Table 3.6 shows the distribution of traceability across different permissions for the 1,758 analysed skills. The permissions are first grouped into Broken, Partial, Complete, with respect to the policies of the skills where these permissions are requested. A total of 2,616 permissions are requested (622 by skills with broken traceability, 485 by skills with partial traceability (75 of these are traceable), and 1,509 by skills that exhibit complete traceability). The most requested permission is the *Device Address* which is requested 648 times by 464 developers. While *Amazon Pay* is the least asked permission requested only 56 times by 33 developers, it tends to be requested more by skills that have complete traceability. In contrast, *Location Services* permission requested by 90 unique developers is found more in skills that exhibit broken traceability.

Table 3.6 Distribution of traceability across different permissions for the 1,758 analyzed skills in the 5 English-speaking marketplaces

Permission	D	R	B		P		C	
			N	%	N	%	N	%
Device Address	464	648	188	29%	130	20%	330	51%
Device Country	330	569	106	19%	77	14%	386	68%
Email Address	251	428	99	23%	77	18%	252	59%
Personal Info.	144	324	67	21%	41	13%	216	67%
Name	173	350	86	25%	82	23%	182	52%
Mobile Number	97	139	36	26%	37	27%	66	47%
Location Services	90	102	35	34%	31	30%	36	35%
Amazon Pay	33	56	5	9%	10	18%	41	73%
Total	1,582	2,616	622	24%	485	19%	1,509	58%
Unique	1,123	1,758	442	25%	306	18%	1,010	57%

D = Developer, **R** = Requested, **B** = Broken, **P** = Partial, **C** = Complete,

3.4.3.3 The Good, the Bad and the Ugly Developers

We next study how many "good", "bad", and "average" developers there are. Table 3.7 shows the number of developers per type of traceability considering the 5 English marketplaces. Overall, we see a total aggregate across markets of 1,730 cases where developers request permissions in their skills, out of which 1,123 are unique developers that post skills in several markets. When looking at unique developers, we see:

The Good: There are 566 developers with all their skills showing complete traceability. All the skills developed by these developers have statements in their privacy policies clearly stating and justifying the permissions they request. This accounts for about 50% of the developers. For instance, developers such as *GoVocal.AI*, *Blutag Inc*, and *Ixartz*

Table 3.7 Developers' disclosure practices (Broken, Partial or Complete) in all 5 English-speaking marketplaces

Markets	Developers	Broken	Partial	Complete	Partial and Complete
US	731	219	145	330	15
UK	402	197	41	146	7
CA	224	88	27	102	2
IN	215	107	23	79	5
AU	202	90	22	82	3
Total	1,730	701	258	739	32
Unique	1,123	350	207	566	21

in the US marketplace have complete traceability between the permissions their skills ask and their privacy policies.

The Bad: There are 350 developers with all their skills broken. This accounts for about 31% of the developers. Skills do not offer an adequate explanation in general when we analyse the partial skills, their privacy statements, and we look at their reviews. One example is *Tagrem Corp* in the US market, with all the skills they developed exhibiting broken traceability, as it does not acknowledge the collection of any personal information in the skills' privacy policy while the skills do request for permissions.

The Ugly: We see 207 developers with all their skills with partial traceability. This accounts for about 18% of the developers. They appear to have a sloppy attitude when writing privacy policies and informing users of how the personal information they request is used. For example, the developer *Geekycoders* with over 30 skills in the US market have partial traceability in all of them. While the developer requests for only one type of permission (First Name), the statement "we do not collect your personal information when you use any part of our products, unless the app specifically lists in any pre-download description" only offers partial traceability since it does not explicitly state what information is collected.

This shows that both average and bad developing practices are commonplace and widespread in Alexa across marketplaces, particularly in the US. Instead of just for individual skills, interventions for developers may also be an effective way of vetting marketplaces.

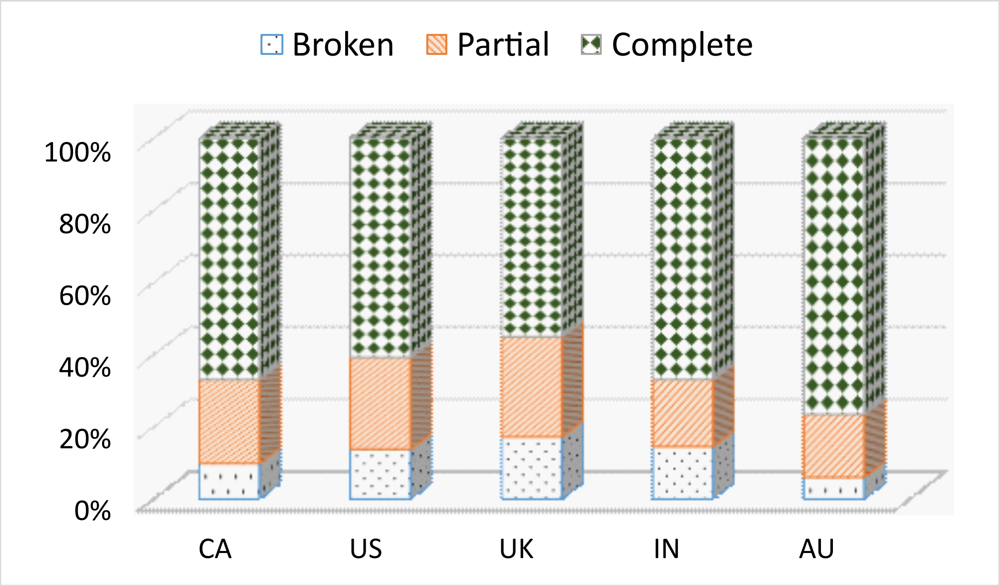


Fig. 3.6 Traceability in skills reusing privacy policies (English-speaking markets). Total: 1498 skills, broken: 196, partial: 358, complete: 944

3.4.3.4 Reused Policies

We observe that some skills are reusing the privacy policies of others. Thus, we measure the prevalence of reused policies and study this practice's impact on traceability by systematically mapping policy links to skills in the same English-speaking marketplace. Figure 3.6 shows an overview of our results, where 175 privacy policy links are reused across marketplaces by 1,498 skills (with the following breakdown[5]: CA-172, US-825, UK-180, IN-172 and AU-149) from 235 developers (CA-37, AU-26, IN-9, UK-39, US-124). Out of all the skills we see reusing policies, only 29% offer a comprehensive policy that adequately describes their practices, 44% show broken traceability between policy documentation and data permissions, while 27% show partial traceability. For example, the skills "Curious City" and "Hits 94" developed by *Voxmatic.io* both with data permission *Device Address* have a privacy policy hosted in a domain that is down (http://voxmatic.io/privacy), though both are fully functional when spoken to.

Interestingly, although many of the skills with the same policy are published by the same developer and collect the same permissions, there are cases where the skill collects a different set of permissions (CA-111, AU-92, IN-109, UK-63, US-548). For example, in the Canadian marketplace, the skills "Big Sky" and "I'm Driving" (both from developer *Philosophical Creations* and with partial traceabilities) use the privacy

[5] The breakdown is read as follows: CA-172 means 172 skills using one of the policies in the Canadian market.

policy link https://driving.big-sky-alexa.com/privacy. However, the first asks from *Alexa Notifications* and *Device Address* permissions while the second asks for *Email Address* and *Mobile Number*. There are also cases where completely different developers still reused the same privacy policy. We see 24 unique instances involving 54 different developers across all of the English-speaking markets. For instance, in the US market, 4 developers, *Evezilla Ltd, Kavson Ltd, Rai Integration Ltd, and Mikk London*, use the same policy link https://www.starfishmint.com/policy/privacy.html.

Finally, another interesting observation is that while some skills reuse broken policies and, therefore, automatically inherit the broken traceability, other skills reusing policies have different traceability due to the different permissions the skill request. For instance, the "Mastering Python Networking Facts" skill by *Network Automation Nerds LLC*, which asks for *Device Country and Postal Code*, exhibits complete traceability with the privacy policy at "https://www.alexa.com/help/privacy". However, the "Stock Price" skill by *JJ* exhibits partial traceability when the same policy is used while asking for the *Lists Read Access* permission, which differs from the other skill's permissions.

3.5 Beyond API Permissions

Amazon enforces access to personal data through a permission model embedded in their APIs, as discussed in Section 3.4.1. However, skills could also request personal data directly from the user without Amazon's API. This can be done via the Web through account linking or via conversations.

3.5.1 Account Linking

Account linking allows users to connect their identity with the one they use in a different system like Google, Amazon, or Facebook [16]. This is implemented in Alexa using OAuth 2.0. A total of 5,230 skills (about 4.7% of the skills in our dataset) are using the account linking feature across marketplaces (with the breakdown: FR-23%, DE-10%, US-6%, UK-6%, AU-4% AU, CA-4%. This is over twice the 2,608 skills (2.3%) that are asking for at least one of the Alexa permissions (see the beginning of Section 3.4). By systematically enabling skills with account linking using a script developed with Selenium [205], we report where they connect to. To do this, we check whether the skill name or developer's name is in the domain used on the account linking URL and/or whether a name of an OAuth service provider is present.

Table 3.8 How skills connect user's Alexa identity with their identity in other system

Account Type	**Number of Skills**	**Percent**
Developer	2,720	52%
Third-Party	1,517	29%
Third-Party or Developer	732	14%
Unresolved	262	5%
Total	5,230	100%

To compile the list of OAuth service providers, we look at 100 skills with account linking features selected at random to note which domains they can connect to. Google, Amazon, Twitter, and Facebook are the only 3rd party OAuth2.0 providers we observed. For completeness, we then include other popular OAuth2.0 providers from the top 50 rank websites on Alexa.com. We are confident that we include most OAuth providers judging by the lower number of unresolved accounts in Table 3.8. The full list of OAuth providers we use is: *Amazon, AOL, Autodesk, Apple, Basecamp, Battle.net, Bitbucket, Bitly, Box, Cloud Foundry, Deutsche Telekom, deviantART, Discord, Dropbox, Facebook, Fitbit, Formstack, Foursquare, GitHub, GitLab, Google, Google App Engine, Huddle, Imgur, Instagram, Intel Cloud Services, Jive Software, kakao, Keycloak, LinkedIn, Microsoft (Hotmail, Windows Live, Messenger, Active Directory, Xbox) NetIQ Okta OpenAM, ORCID, PayPal, Ping, Identity, Pixiv, Reddit, Salesforce.com, Sina, Weibo, Spotify, Stack Exchange, Strava, Stripe, Twitch, Twitter, Viadeo, Vimeo, VK, WeChat, XING, Yahoo, Yammer, Yandex, Yelp, and Zendesk.*

Table 3.8 shows that most of the skills connect to the developer's system, although an important number of skills rely on third-party services such as Google or Facebook for authentication. In particular, 52% of the skills with account linking connect to the developer's site, and 29% to services like Facebook, Twitter, Amazon, or Google, among others. Some skills (14% of all 5,230) allow both, authenticating through a third-party using OAuth before being redirected to the developer's site. Finally, the rest 5% (Unresolved) could not be labelled as developer or third-party. Refer to Table 3.10 for examples of account types used by a sample of skills.

Table 3.9 shows the number of skills with account linking per category (only categories with > 5 skills displayed). We see that Smart Home is the category with the highest number of skills that use the account linking feature. This is natural since it is necessary for smart things like connected cars and smart homes to connect the user's identity with

Table 3.9 Skills with account linking by category (English-speaking markets)

Category	**N**	**%**	**Category**	**N**	**%**
Smart Home	2212	42.29%	News	57	1.09
Business Finance	384	7.34%	Utilities	56	1.07%
Music Audio	341	6.52%	Home Services	36	0.69
Uncategorized	273	5.22%	Movies TV	23	0.44
Lifestyle	249	4.76%	Novelty Humor	23	0.44
Productivity	241	4.61%	Sports	19	0.36
Health Fitness	226	4.32%	Organizers Assistants	18	0.34
Games Trivia	179	3.42%	kids	16	0.31
Shopping	150	2.87%	Public Transportation	16	0.31
Education Reference	148	2.83%	Knowledge Trivia	13	0.25
Food Drink	127	2.43%	Streaming Services	13	0.25
Social	98	1.87%	Local	10	0.19
Travel Transportation	94	1.80%	Calendars Reminders	8	0.15
Connected Car	79	1.51%	Navigation Trip Planners	7	0.13
Weather	59	1.13%	Cooking Recipes	6	0.11

their identity in the developer's system. There are, however, other categories with many skills with account liking too, such as Music & Audio and Business & Finance, as these naturally aim to allow the user to connect to their existing online resources through a voice interface.

Table 3.10 shows some of the skills in the Indian market with account linking and the type of account they connect to. We can see that skills like "GoToMeeting for Alexa", "Crypto Genie", and "Uber" can connect the identity of users with their identity on the developer's system. While skills such as "Bollywood Mania" can only link the user's identity with a third-party system, skills like "HiCare" can connect the identity of users with their identity on either the developer's system or third-party system.

We then look at the traceability of the skills with account linking and at least one permission. Table 3.11 shows that 47% of them have broken or partial traceability. While these broken and partial results are conclusive, when it comes to *complete* traceability, results may not be conclusive — recall that account linking could enable the collection

Table 3.10 Sample skills and account linking domain (India)

Skills	**Account Type**
GoToMeeting for Alexa, Crypto Genie, J.P. Morgan, Nurturey maths, AGL, Voice Prototypes, Sayspring, Zomato, JioSaavn, Ola, I'm Driving, Phone Genie, Uber, Vodafone	Developer
paisabazaar, StockInvest, Escape the Room, Brightidea Home, Voice Rewards Me, Commvault,	Amazon
Starfish Local	Google, Amazon
Bing Bong	Facebook, Google
Trivia Monster, The Dark Citadel	Facebook, Google, Amazon, Developer
Bollywood Mania	Twitter
BBC Good Food, cure.fit, Fitbit, KEYCO air	Facebook, Google, Developer
HiCare	Facebook, Google, twitter, Developer

of further personal data without the need to use data permissions (i.e., taking the user out of Alexa). Accounting for the exact data collected by developers through account linking is challenging because: 1) third-party sites used for account linking have all different formats and are thus challenging to scrape; 2) developers may ask for any type of personal information (at any point, and not just during the registration).

To study further the process of account linking and the data involved, we randomly selected 70 skills and performed the account linking process manually. We see 65 skills (92.85%) requesting personal data without using the Alexa API. We next discuss some case studies we find as a result of this analysis. First, we look at "GCSE Revision", a skill developed by *Palm UK Ltd* that helps users to prepare for exam-board specific GCSE (General Certificate of Secondary Education) science questions. While the skill asks for permissions *Full Name* and *Email Address*, we see that during the linking process, the skill asks, in addition, for the user's *address*. This is particularly problematic as the skill is effectively bypassing Alexa's permission system, using account linking as a proxy to obtain additional data from users. This also shows a prevalent issue we have discussed earlier (see Section 3.4.3): this skill targets students between 12 and 16 year old students. Since GCSE has access to the Full Name, the skill could have been

Table 3.11 Traceability results for English-speaking markets skills with account linking and at least one permission

Market	Skills	Broken		Partial		Complete	
		N	%	N	%	N	%
US	133	30	22.6%	35	26.3%	68	51.1%
UK	46	6	13.0%	13	28.3%	27	58.7%
CA	30	1	3.3%	6	20.0%	23	76.7%
AU	27	2	7.4%	4	14.8%	21	77.8%
IN	26	5	19.2%	5	19.2%	16	61.5%
Total	262	44	16.8%	63	24.0%	155	59.2%
Unique	164	39	23.8%	38	23.2%	87	53.1%

collecting the family name of children in the UK that are preparing for the test, together with how well they perform. Note that children's last names can be usually inferred even if the Amazon account was under their parents' names. Therefore, the developers of GCSE could send a targeted postal advertisement to the household. Finally, we also look at the "DaddySays" skill by *Tagrem Corp* in the Kids category. The skill collects the email address of the users during the account linking process in addition to the information collected through the permissions *Lists Read Access* and *Lists Write Access.*

3.5.2 Collection via Conversation

We interact with 100 randomly-chosen skills from those that do not request permissions using the Amazon API and that do not use the account linking feature either. We see that 35 are single-interaction skills (they just provide an answer to the user without further interaction), 45 are conversational skills, and the remaining 20 return no response or responded with an error (e.g., "Sorry; I am not sure about that"). Out of the 45 conversational skills, we find *3 skills* asking for personal information directly when conversing with users, effectively bypassing the Amazon model for data collection practices. These are "Would You Rather for Family" by *Voice Games*, "The Bartender" by *Pylon ai*, and "Phone Tracker" by *S4 Technology, Inc.* For instance, the latter asks for a phone number as part of the conversation. A systematic, more in-depth study of personal information via conversation is therefore presented in Section 4.4, after we present an automated tool to facilitate this study, which would be impractical to conduct at this stage.

3.6 Discussion

3.6.1 Key Findings and Limitations

3.6.1.1 Potential Privacy Violations

We see bad privacy practices in about 748 skills (43%) of those that request permissions in English-speaking marketplaces — recall English-speaking skills are 82.7% of the total, involving 557 (50%) of the developers with skills that request permissions. Although it could be that some developers have multiple Amazon developer accounts and the real number of entities behind those accounts are less, this still paints a very worrying picture. Particularly worrying cases are those in categories like Kids, Schools, and Education, which exhibit broken (72 skills in total) and partial (17 skills in total) traceability. Beyond the traceability itself, this may have other implications, as skills in those categories might be subject to even tighter privacy regulations for children, such as the US Children's Online Privacy Protection Act (COPPA) of 1998 [51].

Other categories, such as the Utilities category, also have countless examples of an unjustified collection of personal information i.e., with broken traceability. Skills, such as "Mock Interview" *by Graylogic Technologies*, ask for permissions — like *Device Address* — without acknowledging such collections in its privacy document. Furthermore, we even see a case of a skill collecting information while stating that it is not currently used. This is the case of the "Bin reminder" skill by *Shane* that collects the user's *Device Address.*

3.6.1.2 Bulk Skill Creation

There are a few developers that publish hundreds and even thousands of skills (cf. Section 3.3). We see some developers using automated systems to develop and/or deploy skills. This democratises the creation of skills, but also lowers down the entry barriers for criminals and can foster the commoditisation of unwanted skills. Previous works have recently shown that miscreants leverage online app generators to publish unwanted apps in Android [126]. We believe that Amazon Alexa will suffer from the same challenges as Google or Apple with the vetting of apps in Android or iOS markets [97]. In fact, recent research seems to point toward issues with the current vetting process in Alexa [44]. We posit that the research community needs to develop detection mechanisms tailored to the SPA ecosystem. However, one important limitation is the lack of access to the skill software (e.g. deployed in the cloud), which also constrains our analysis.

3.6.1.3 Potential for Impersonation and Sybil Attacks

We see thousands of skills from different developers with a similar name (impersonation) and hundreds of skills from the same developers in the same market (Sybils). We have seen how this may pose a threat to the invocation system of SPAs. Alexa has recently introduced a mechanism to provide name-free skill interaction, i.e., a skill can be invoked without its skill invocation name. For this, Amazon uses rich contextual information to select the best skill that matches the user request [116]. This widens the attack surface as impersonation attacks may not necessarily need to craft skills with the exact same name. While our work does not consider name-free skill in our impersonation analysis, our findings can be seen as an under-approximation of the problem, suggesting that motivated attackers could take advantage of this.

3.6.1.4 Market Migration

Amazon allows developers to publish the same skill in different markets. Our study shows that there is an important overlap of skills across markets, which includes the use of the exact same privacy policy. This has important regulatory implications derived from the different local regulations in a globalised ecosystem, similar to what we have seen for cookies in the wake of GDPR [98, 102], and the different languages across marketplaces. Note, however, that skills that are pushed from English-speaking markets to other markets now go through a language migration process, and privacy policies may differ for the same skill [10]. This process was not in place at the time we started our study.

3.6.1.5 Advertisement

We see evidence of skills embedding advertisement as part of their responses. A good example is the “myTuner Radio Player Canada” skill by *Appgeneration Software technologies* where a full-screen ad pops up (in SPA devices with a screen) when the user selects a different station. Another example is the “Sleep and Relaxation Sounds” skill by *Voice Apps, LLC* which keeps advertising its premium subscription and overwhelms users with constant commercials until they buy its premium service.

3.6.1.6 Spamming

We observe several skills spamming users for reviews, like "Sleep Sounds: Ocean Sounds" by *Invoked Apps LLC* and "Good Morning Gorgeous" by *Skillex Studios.* Also, there are skills such as "Hits 1 Latina" by *autopo.st*, and "Sleep and Relaxation Sounds" by *Voice Apps, LLC* that do not respond to *Alexa stop command* and keep spamming users with unsolicited information after invocation. "Night Light" by *labworks.io ltd*, requires users to say *"Alexa stop nightlight"* instead of the usual stop command. "Sleep Sounds: Pink Noise" by *Voice Apps, LLC* both spams users for a review and fails to respond to Alexa stop command.

3.6.2 Developers' Business Models

Our key findings warrant a further discussion about the motivations developers may have to develop skills. In this regard, our study identifies over 47K developers that have contributed to the Alexa marketplace with a rich ecosystem of skills (cf. Table 3.1 in Section 3.3). Since Amazon forbids advertisements in skills, an important open question is what do these developers gain and what are their motivations. First, companies may be interested in offering a voice-over interaction with the user through Alexa. Examples are skills in the Travel & Transportation category (e.g. to order a ride) or in the Food & Drink category (e.g. to order a pizza). In these situations, developers need to bind the user's Amazon account with the external service via account linking, as mentioned before. Interestingly, we only observe 5% of the skills requesting account linking, which is circa 5K unique skills (cf. Section 3.5.1). We can, thus, conclude that account linking is not yet the primary motivation for most developers.

Amazon promotes in-skill purchasing and paid subscriptions, which allows developers to sell premium content in skills such as extra features, in-game elements and interactive stories. This skill content can be offered as: i) subscriptions, where developers charge users recurrently to access premium features, ii) one-time purchases, where users pay once to have permanent access to the premium features and content, and iii) consumables, where content can be purchased, exhausted and purchased again (e.g., extra-lives in games [17]). In-skill purchasing is not yet supported by all markets. At crawling time, we only see 1,535 (1.37%) skills with in-skill purchasing or paid subscriptions in the US and UK. Prices for in-skill products range considerably, from 0.99 to 99 USD/GBP. "Sleep Sounds: Hair Dryer", "Sleep Sounds: Harp Sounds", and "Sleep Sounds: Heavy Rain" developed by *Voice Apps, LLC.* are examples supporting in-skill purchases. However, judging by the number of skills with in-skill purchasing, we can conclude that this is not the main source of income yet. Developers may also offer paid skills in the future, but this is not an option at writing time.

Overall, we see that most of the skills in our dataset are free (both to enable and to use fully). The total number of skills that are free and do not have in-skill purchasing is 110,494 (98.6%). Out of these, 106,289 (94.9%) do not offer account linking either. Therefore, though there might be small incentives, such as the Amazon incentive program [12], which rewards skill creation (e.g. with a smart plug), or the simple curiosity to develop in a new environment, we conclude that *there is an ample number of skill developers for which there is no clear business model.* Some of these skills might be in the data monetisation business. An obvious example is "Pixated Salat", the prayer skill developed by a large advertising company (c.f., Section 3.4.3). Another example is the "Autochartist" skill, which states in their privacy policy that the data they collect may be used "to provide you with news, special offers and general information about other goods, services and events which we offer that are similar to those that you have already purchased or enquired about unless you have opted not to receive such information". It is, however, unclear how users of skills can opt-out.

3.6.3 Mitigation

Tackling bad practices from third-party developers is a challenge to systems security. Having a well-defined — fine-grained — permission model is an important step forward. However, we have seen that: 1) developers can bypass it, and 2) developers do not offer transparency about the way they utilise users' information. We next present a range of countermeasures that could mitigate the risk of enabling third-party developers access to users' data. First, Amazon should not allow developers to reuse policies verbatim as this is error-prone, as we have seen. Second, both Amazon and the developers should thoroughly review the traceability themselves. Such review needs to be a continuous routine for effectiveness. However, since the number of skills is large and new skills are added to the ecosystem daily, it will be time-consuming to analyse traceability manually. Hence, there is a need to automate the analysis at scale to help achieve consistent, persistent, and repeatable measurements.

Third, publishing the code of the skills or the binaries will considerably foster research in the area, similar to what we have witnessed in Android [42]. While approaches to detect unwanted skills can be adopted from Android, the Alexa ecosystem has unique features like voice recognition that present novel challenges (e.g., voice masquerading attack [252] — malicious skills that pretend to hand over the control to another skill).

Finally, another important mitigation is to increase user awareness, as users are known to have incorrect or, at best incomplete mental models of how assistants such as Alexa work [231, 1]. This may be even more challenging when users interact with skills aside of the policy enforcer, such as during the account linking process or with conversational skills. There have been works to defend users against malicious activities on online services (e.g., spam [216], or phishing detection [74, 64]). Solutions in this direction will have to consider that the nature of assistants brings, again, unique challenges.

3.6.4 Responsible Disclosure

To enhance the security of the skill ecosystem, we safely report our findings: We perform a responsible disclosure process, starting from mid-August 2020, as follows. First, we notify all skill developers who are not engaging in good data practices whenever we have their contact details. Second, we also report our findings to Amazon and have confirmed that the skill store team has taken action. All in all, we reported 675 skills with privacy issues to Amazon and the affected developers.

3.7 Conclusion

The Amazon skill ecosystem has grown rapidly without a clear business model. We have presented a systematic measurement study that provides a large-scale analysis of this ecosystem, analysing over 199k third-party skills. By looking at the distribution of skills and the diversity of developers, we have shown current practices in the wild and offered a unique understanding of developers' motivations. We have studied the permission model Amazon offers to developers and investigates how skills use permissions to collect personal data. We also designed a methodology that identifies potential privacy issues by analysing the traceability between the permissions and the data practices stated by developers. While transparency is paramount to let users make informed decisions about the disclosure of their data, our result shows that 43% of skills do not comprehensively disclose their data practices. Furthermore, we have also seen how skills may bypass Alexa's permission system by requesting personal information without using their APIs (e.g., on an external domain via account linking). This indicates that even skills with a complete privacy policy can pose a risk to users. We have discussed the most concerning practices and the implications of our findings.

As highlighted in Section 3.6.3, there is a need to automate traceability analysis at scale to have a continuous and repeatable measurement. However, one interesting question that naturally flows from this discussion is *how can we do this in an automated way at scale?* In the next chapter, we propose a highly accurate system based on machine learning and natural language processing to help automate the traceability analysis at scale.

Chapter 4

Automated Traceability Analyser

The number of skills is large and new skills are added to the ecosystem daily, making it challenging to perform thorough traceability analysis manually. Besides, it is important to achieve consistent, persistent, and repeatable traceability measurements. To help scale the traceability analysis measurements, we need an automated traceability analysis tool. This chapter proposes a system, SkillVet, based on machine learning and natural language processing that can help automate traceability analysis at scale. SkillVet identifies relevant policy statements and assesses the traceability as complete, partial, or broken. We accomplished the remaining part of OBJ2 in this chapter as presented in Section 1.2: *To develop a practical method for performing traceability at scale.*

4.1 Introduction

As was mentioned in the previous chapter, one way to mitigate the risks associated with third-party developers accessing users' data is by conducting a thorough traceability review. This will help identify skills with bad privacy practices. However, performing such a check manually at scale will be time-consuming due to the increasingly large amount of skills. Hence, there is a need for an automated approach to help scale this analysis. Automating traceability analysis requires an automated tool that can perform repeatable and consistent measurements. The tool should, in particular, be able to:

1. Recognise the unique features of SPA — for example, the lack of the skills' code or executable files.
2. Perform accurate analysis across the different traceability types and the various permissions.

3. Support sentences describing more than one permission, and
4. Consider only those permissions that have privacy implications and need to be disclosed in the privacy policy.

In this chapter, we present a system, SkillVet, based on machine learning and natural language processing that can help automate traceability analysis at scale. In addition, SkillVet could detect skills with inappropriate usage of data permissions at large and help protect users' privacy. Our evaluation dataset of 972 *unseen* skills and their policies shows that SkillVet could achieve 99% accuracy in identifying broken policies and 93% overall accuracy.

4.2 SkillVet Architecture

Given a skill, SkillVet first identifies and classifies all statements in its privacy policy that relate to data practices over personal information. It maps each data statement with one or more Alexa permissions. These are the permissions the skill justifies in the privacy policy. SkillVet then compares these permissions with those the skill is authorised to request through the Amazon API during runtime. Depending if the permissions requested match those found in the policy, the skill is then classified as having a *complete*, *partial*, or *broken* privacy policy.

Figure 4.1 presents an overview of the SkillVet system. The automated traceability analysis system consists of two parts, the *Sentence Classifier* and the *Traceability Analyser*. The Sentence Classifier (Section 4.2.1) spawns several models that perform the mapping mentioned above. The Traceability Analyser (Section 4.2.2) collects all data permissions found in the privacy policy and compares them with the original set of data permissions requested by the skill. This is done to determine the traceability between the use of permission and its justification in the policy.

4.2.1 Sentence Classifier

To identify the privacy permissions requested in each sentence automatically, we manually created the *permission-by-sentence dataset (PBSD).* The PBSD compiles 10,409 annotated sentences from 532 original Alexa policies randomly chosen from the TBPD described in Section 3.4 (the rest of TBPD is used as *unseen* data for the evaluation as detailed in Section 4.3.2). The annotations are tags that associate each statement to one (or more) of the Alexa permission categories of interest. That is, PBSD contains

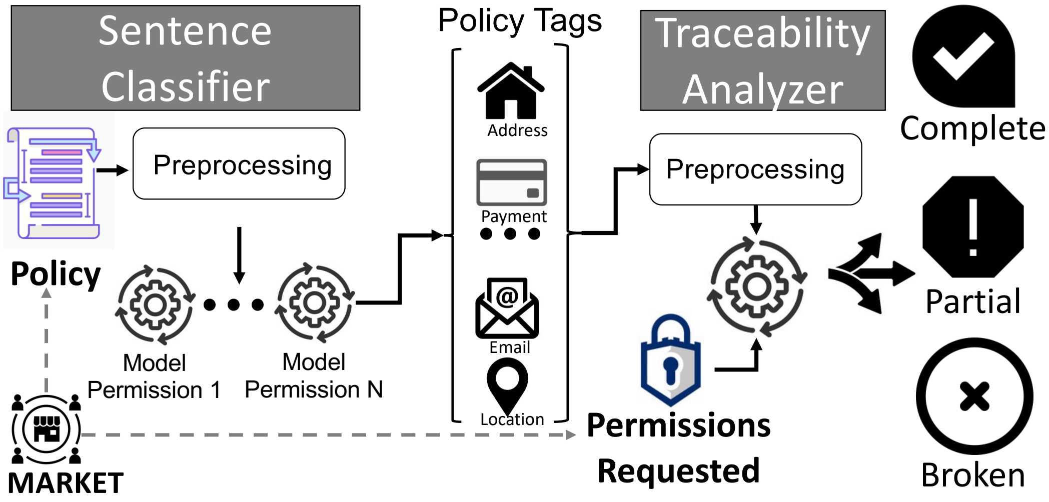


Fig. 4.1 Overview of SkillVet

sentences with permissions needed for the traceability analysis as discussed in detail in Section 3.4.2 (*Amazon Pay, Device Address, Device country and postal code, Email Address, Location Services, Mobile Number, Name, and Personal Information*) and the negative label, named *None*, which includes sentences that do not belong to any of the Alexa permission classes considered.

The distribution of tagged sentences per permission category in the PBSD is presented in Table 4.1. Note the uneven distribution of sentences among classes. The permissions with the largest number of statements are "Personal Information" and "Name". Although we strive to have representative statements for all categories in Alexa, some less actionable permissions such as *Device Address* or other recently incorporated permissions like *Amazon Pay* are still not commonly used among skills. Therefore, they are not easy to find and tag. To mitigate this issue, we extended the PBSD with sentences from the 350 Android privacy policies (APP-350) [256] only for permissions that map with Alexa's data permission categories, i.e.: the *Device Address*, *Device country and postal code*, *Email Address*, *Location Services*, and *Mobile Number*.

The problem we tackle is a multi-label classification problem, as sentences can be referring to more than one permission. For instance, the sentence "We collect your name, and postal address" indicates the collection of permissions *Name* and *Device country and postal code*. One well-known way to tackle multi-label classification problems is to

Table 4.1 Sentence count per permission in PBSD and the extended PBSD+APP-350

Alexa Data Permission	#PBSD	#PBSD+APP-350
Amazon Pay	135	135
Device Address	254	776
Device Country and Postal Code	244	518
Email Address	205	2181
Location Services	186	533
Mobile Number	97	941
Name	416	416
List Access	2,482	2,482
None	6,390	6,390
Total	10,409	14,372

transform the problem into a number of binary classification problems.[1] In particular, the sentence classifier is formed by 9 binary models, one per each of the 8 Alexa data permission categories we model, and one for the sentences that do not belong to any permission category. All binary classifiers are trained following a one-vs-all strategy. For every classifier modelling a particular permission, we label all sentences relating to that permission as the positive class and all other sentences as the negative one. Given a sentence from a privacy policy, each of the classifiers then evaluates whether that sentence belongs to the permission category being tested or not. Therefore, we obtain a multi-label classification of the sentence. That is, every classifier makes an independent assessment, and as a result, we are able to model sentences describing more than one permission.

4.2.2 Traceability Analyzer

The traceability analyser collects the data permissions outputted by the Sentence Classifier and compares them with the permissions requested by the skill in two steps: policy preprocessing and traceability check.

[1] Note that other transformations may be possible [141], e.g., transforming the multi-label classification problem into a multi-class classification problem but that would require 2^8 classes.

4.2.2.1 Policy Preprocessing

Given an Alexa policy, the Traceability Analyser first splits the policy into sentences and preprocesses each of the sentences in the same way as presented in Section 3.4.2. After this, it removes sentences non-related to data permissions, such as those in which the authors provide some contact details, e.g.: "if you have any inquiries about this skill, please contact us by email". These sentences are often structured similarly in all policies, use similar terms, and are usually wrongly assessed by classifiers, which are unable to identify that the sentence does not describe a data permission request. To detect this special set of sentences, we use a keyword blacklist containing words such as 'contact us' or 'call us'. The same is done with negation terms such as "does not", or "doesn't". This step filters out negative sentences such as "this skill does not need your email address", which are quite common among skill policies. If a sentence contains any of these blacklisted keywords, the sentence is ignored and not classified, as modelling negative statements is important to determine the correct meaning of a privacy policy [23].

4.2.2.2 Traceability Check

After the policy preprocessing, the Traceability Analyser runs each of the 9 classifiers through all the remaining sentences, collecting all data permissions in the policy. Note that the None classifier prevails over the others to handle contradictions in data policies [23]. That is, we consider that a policy does not have a (proper) data statement if there are permissions associated with a policy in addition to a positive classification of the None classifier. Next, through the comparison between permissions requested by the skill and those automatically found in its policy, the Traceability Analyser classifies the traceability of the skill as *broken*, *partial* or *complete*, in the same way as in Section 3.4.

4.3 Implementation and Evaluation

In this section, we describe how we train our sentence classifiers and discuss the system's performance.

4.3.1 Sentence Classifier Training and Validation

We hypothesise Support Vector Machines (SVM) to work better than Random Forest (RF), while deep learning not being applicable as we do not have enough data. SVM is suited for binary classification problems [55]. SVM also usually performs better for NLP tasks with n-grams than other classifiers [3]. However, we also tried classification using RF, as shown later.

Each of the 9 models is a binary SVM classifier built on top of an n-gram binary vectoriser and a tf-idf layer. Note that approaches such as sentence embeddings [171] could be used, but the repetition of similar constructs among policies indicates that a simpler, n-gram classifier is also appropriate. We test various parameters over a set of 5-fold cross-validation runs for each of the 9 models, including the size of the n-grams, the SVM loss method, the SVM alpha value, and different oversampling and undersampling strategies to balance the classes of the *permission-by-sentence* dataset (PBSD). We make sure that each data permission is represented consistently among the instances selected for training and testing in order to increase the robustness of the classifiers against sentences of all types. The parameters that consistently returned better F1 and accuracy scores over the different 5-fold cross-validation are then used to train with all data.

In a nutshell, the best parameters are: we use n-grams of size 1, 2 and 3, stratified random undersampling and oversampling as balancing strategy, *modified huber loss* as the SVM loss function, and an *svm alpha* of 10^{-5} for the SVM classifier. After undersampling and oversampling certain classes, on average, each classifier was trained and tested using between 2K and 8K of the total sentences found in the *permission-by-sentence* dataset. Interestingly, we observe that the classifiers perform best when we use *at the same time* n-grams of different sizes (1,2,3).

Table 4.2 K-fold validation for the sentence classifier using SVM

Alexa Data Permission	**F1**	**Accuracy**
Amazon Pay	0.987	0.991
Device Address	0.919	0.908
Device Country and Postal Code	0.921	0.909
Email Address	0.967	0.965
Location Services	0.960	0.973
Mobile Number	0.978	0.977
Name	0.977	0.983
List Access	0.986	0.982
None	0.993	0.991

The final versions of all binary classifiers that are deployed in SkillVet are trained using 100% of the PBSD dataset and then evaluated using a set of extra *unseen* 523 sentences to check their performance at a sentence-permission level (see below for the evaluation of SkillVet as a whole). The average F1-score and accuracy metrics obtained are reported in Table 4.2. Note how the sentence classifiers are able to differentiate and classify each of the permissions correctly, obtaining F1-scores and accuracy of over 0.9 for all data permissions.

When selecting the choice of the machine learning algorithm, it is interesting to note that we also performed training and classification using Decisions Tree (DT) and RF. However, we discovered that SVM offered the best results overall. For instance, Table 4.3 shows the F1-score and accuracy metrics of each of the permissions using RF classification. We see that the scores obtained using SVM outperform that of RF except for *Device Address* and *Device Country and Postcode* permissions. Existing work [3] has corroborated this finding that SVM outperforms other traditional Machine Learning algorithms when used for Natural Language Processing tasks with n-grams.

Table 4.3 K-fold validation for the sentence classifier using Random Forest

Alexa Data Permission	**F1**	**Accuracy**
Amazon Pay	0.975	0.983
Device Address	0.974	0.987
Device Country and Postal Code	0.967	0.977
Email Address	0.793	0.878
Location Services	0.851	0.948
Mobile Number	0.963	0.975
Name	0.925	0.968
List Access	0.968	0.968
None	0.969	0.969

4.3.2 SkillVet Evaluation

To evaluate the performance of SkillVet as a whole, we compare the traceability results we obtained with the *unseen* remaining subset of 972 skills and their policies from the *traceability-by-policy dataset (TBPD)*.[2] That is, the policies of these skills are not used to create the PBSD as detailed in Section 4.2.1, so they are not used at any point for training, validation, fine-tuning, or evaluation of the sentence classifiers. It takes SkillVet 7 mins 23 secs to process all 972 skills (an average of 0.45 secs per skill). As shown in Table 4.4, SkillVet is able to correctly classify 93.1% (905) of the previously unseen 972 policies. The highest accuracy measures are obtained when identifying broken and complete policies (98.5% and 93.9% accuracy), while partial traceability seems to work comparatively worse, with a few miss-classified as complete.

Table 4.4 Confusion matrix comparing SkillVet with a human analysis for the 972 *unseen* skills from TBPD

SkillVet	Actual Traceability on 972 unseen policies			
	Broken	Partial	Complete	Total
Broken	264 (98.5%)	8 (4.9%)	11 (2.0%)	283
Partial	0 (0.0%)	134 (81.7%)	22 (4.1%)	156
Complete	4 (1.5%)	22 (13.4%)	507 (93.9%)	533
Total	268	164	540	972

We further investigate the reasons behind the 67 errors in total (out of 972 skills), which we organise into the following three main types:

1. *Class error* (32 cases, 48%) occurs when the policy is wrongly analysed due to an error in one of the sentence classifiers producing the wrong class

2. *Fair error* (15 cases, 22%) occurs when even human struggles to identify the correct class because the sentence can be interpreted in different ways, e.g., the most common cases are sentences that refer to *address* but it is not clear whether it is the *device address* or the *email address*, in fact, the manual analysis for traceability would consider this as partial precisely because it is not exactly clear what address the sentence is referring to

3. *Filter error* (20 cases, 30%) is when the error is attributed to the preprocessing stage of SkillVet, e.g., most of the cases are due to sentences where there is a

[2] Note that we did not consider a further 254 skills in TBPD, because their traceability is trivial: they have missing policies, dead links, or empty policies. A simple check without sentence classification is enough to assess them broken. Including them would potentially bias the evaluation away from the more complex, error-prone cases.

complex contradiction, in the sense that there is a negation but associated to a positive non-collection action, actually suggesting a good practice (e.g., not shared with, not disclosed to) in an overall positive sentence

We next discuss how one might be able to improve the performance of SkillVet further in light of the errors. For Class errors, one way to improve would be to have more tagged sentences in the PBSD. In general, however, SkillVet shows a very good performance, considering that PBSD has less than 250 instances for most of the permissions (cf. Table 4.1). For Fair errors, this is more challenging, but one possible improvement could consider a separate sentence classifier trained with cases where it is not clear what address is being considered, where a human looking at the wider context can make the right assessment (though in most cases as stated above, the human tagger may not be able to ascertain this either). Finally, for Filter errors, there may be room for improvement by considering a more sophisticated preprocessing approach, perhaps aided with an ontology, similarly to [23]. However, this may also introduce errors in turn, and the way SkillVet accounts for negations and contradictions seems effective in general for this domain.

4.4 Use Case of SkillVet

Data collection via conversation allows developers to bypass Alexa's permission system effectively. In this section, we demonstrate how SkillVet could be used to perform traceability analysis for collection via conversation. This is one particular issue detected in Chapter 3 where an automated approach is more helpful than manual inspection.

We used our system, SkillVet, in tandem with recent work, SkillExplorer [88], which has provided an initial method to discover skills collecting personal data via conversation. In particular, We studied the dataset of 100 skills (developed by 89 distinct developers), collecting personal information via conversation provided by the authors of SkillExplorer [88]. Out of all these skills, 82 collect personal information only through conversation, 11 collect personal information via Alexa API in addition to the one collected during a conversation, and 7 also use account linking.

Table 4.5 summarises the collection method used and shows the results of the traceability analysis. Overall, we see that 35% of the skills exhibit broken traceability. In this case, all issues are attributed to personal data collected alone during a conversation. We also see that 20% partially disclose their data practices in their privacy policy, and the remaining 45% exhibit complete traceability. One good example of a broken skill is the “Praise Me” skill by *Jackson Jacob* with skill id B07G5B4P7R and a customer rating of 4.8 out of 5 in the US market. This skill is available across all five English-speaking countries. According to its description, this skill is supposed to praise the user after asking for their name. However, the skill not only requests the user’s name during the conversation without declaring the permission for it, but it also fails to disclose its data practices in a privacy policy. Furthermore, the skill has a privacy policy link that takes the user to a porn site. There are also two developers with skills having different traceability outputs. *Voice first tech* has skills that exhibit partial and complete traceability (1 skill in each category), while *Volley inc.* has broken and complete traceability skills (1 broken, 2 complete skills).

Table 4.5 Traceability result with SkillVet on subset of skills found in [88] collecting personal information via conversation

How PII were collected	Traceability			Total	
	Broken	Partial	Complete	Skills	Devs
Conversation + Account Linking	—	1	6	7	5
Conversation + Alexa API	—	5	6	11	9
Conversation Only	35	14	33	82	79
Total number of **Skills**	35	20	45	100	—
Total number of **Developers**	35	16	40	–	89

Table 4.6 Confusion matrix comparing SkillVet with a human-centred analysis for the set of the skills given by [88]

SkillVet	Actual Traceability			
	Broken	Partial	Complete	Total
Broken	33 (94.3%)	2 (8.7%)	0 (0%)	35
Partial	1 (2.9%)	17 (73.9%)	2(4.8%)	20
Complete	1 (2.9%)	4 (17.4%)	40 (95.2%)	45
Total	35	23	42	100

We also compare the results given by SkillVet with a human-based traceability analysis made by one of the authors for the 100 skills. As shown in Table 4.6, SkillVet achieves very good accuracy, being able to identify the traceability for 90 of the 100 skills correctly. It is particularly very good at spotting complete (95.2%) and broken (94.3%) traceability and good for partial (73.9%). Of the 10 errors in total, 8 are Class errors, 1 Filter error and 1 Fair error. These results align with what we report for the unseen 972 skills that request permissions through Alexa API in Section 4.3.2.

4.5 Conclusion

In this chapter, we have presented SkillVet, an automated traceability analyser based on machine learning and natural language processing. SkillVet can systematically review traceability between the data practices specified in privacy policies and the data permissions. The system overcomes challenges in modelling privacy policies, such as contradictions like negations and statements related to multiple types of personal information. We tackled a multi-label classification problem, as sentences can refer to more than one permission, by transforming the multi-label classification problem into a sequence of binary classifications. Every classifier makes an independent assessment, ensuring we successfully model sentences describing more than one permission. SkillVet can correctly differentiate and classify each of the permissions correctly, obtaining F1 and accuracy scores of over 90% for all data permissions and overall accuracy of 93%. While self-contradictions are orthogonal to the traceability analysis we do, SkillVet accounts for negations in a way proven effective for traceability analysis.

Unlike the study in [134] where their automated method that uses PoliCheck [24] underneath has 83.3% precision when detecting complete policies, SkillVet has a precision of 94%. This better precision could be attributed, among others, to SkillVet covering all data types in Alexa (incl. Lists and Amazon Pay), while in [134] the ontology they use for PoliCheck only considers a subset of the data types. Further, PoliCheck was trained *only* with android privacy policies, while SkillVet was trained with Alexa skill privacy policies, only complemented with a few android privacy policies.

One of the key findings presented in Chapter 3 is that around 43% of the skills (involving 50% of the developers) that request permissions follow bad privacy practices. This finding raises a number of interesting questions that deserve further attention. This includes questions such as how responsible disclosure helps improve skills privacy practices and whether the practices are getting worse or bad as time goes by (including if there is a change in the effectiveness of providers spotting potential issues). Addressing these essential questions may help understand how to improve these bad privacy practices. In the next chapter, we investigate how the data practices of SPA skills has evolved from 2019 to 2021 and identify the key factors that influenced any changes, which could be crucial for improving the privacy practices of SPA skills.

Chapter 5

Measuring Alexa Skill Privacy Practices across Years

To suggest better ways of improving the privacy practices of SPA skills, we need to understand how the skills data collection practice has changed over time and identify what influences these changes. In this chapter, we perform a systematic and longitudinal measurement study of the Alexa marketplace. We shed light on how this ecosystem evolves using data collected across three years between 2019 and 2021. We leverage SkillVet to demystify developers' data disclosure practices and present an overview of the third-party ecosystem. We see how the research community continuously contribute to the market's sanitation, but the Amazon vetting process still requires significant improvement. We measure the effect of the responsible disclosure we did in Chapter 3, where we reported 675 skills with privacy issues to Amazon and the affected developers, out of which 246 skills suffer from important issues (i.e., broken traceability). We see that 107 out of the 246 (43.5%) skills continue to display broken traceability almost one year after being reported. As a result, the overall state of affairs has improved in the ecosystem over the years. Yet, newly submitted skills and unresolved known issues pose an endemic risk. This chapter fulfils OBJ3: *To perform a longitudinal measurement study of the SPA skills and explore how the different concerning privacy practices permeate through the markets.*

5.1 Introduction

Along with the growth in the number of skills, there is increasing concern over what risks third-party skills may pose to users [1, 142, 2, 183, 152]. Skills widen the attack surface of SPA as malicious actors may develop potentially harmful software that could affect the security and privacy of the users. Recent studies have been looking at the various issues in the use of third-party skills, including publishing potentially harmful skills [252, 127, 206], performing unjustified data collection practices, covertly eavesdropping on conversations [88], or skills performing voice squatting attacks (purposely invoking services under a name that sounds like a popular skill but is spelt differently to hijack its invocation) [134].

Although previous studies have delved into the different attack vectors inherited from using third-party skills, it is unclear to what extent current attacks permeate through the markets. Lessons learned from other platforms like smartphones [25, 78] indicate that SPA operators will struggle to keep the pace in the fight against misbehaving skills. This prompts us with the following open question: how effective are SPA market operators in helping protect users? One key feature of SPA that needs to be considered when answering this question is the strong dependency they hold with the cloud. Skills can host their services on remote systems that are external to the SPA provider. This makes it easy for developers to modify the functionality of the skill after its publication.

To usher how SPA markets are protected in a drifting landscape, it is imperative to study to evaluate the effectiveness of existing measures against malicious threat actors over time. This work is the *first* to measure the changes in Alexa skill developer privacy practices over time, our measurement ranging from 2019 to 2021. We focus on the following research sub-questions:

- RQ5.1 — Has the overall state of affairs regarding data practices in the third-party skill ecosystem improved over time? (Section 5.3)
- R5.2 — Is the collection of personal information explained better nowadays? (Section 5.5)
- RQ5.3 — What influence changes over time and has there been an improvement in the review and certification process? (Section 5.6)
- RQ5.4 — Are skills effectively bypassing the permissions system? (Section 5.7)

To answer these questions, we design a methodology in the next section to perform a data practice measurement, which offers an independent assessment of the skill marketplace.

5.2 Our Measurement Methodology

As illustrated in Figure 5.1, we use the Web scraper to collect data from the Amazon Alexa marketplace at different points in time. We refer readers to Section 3.2 for a detailed breakdown of the collection method and strategy. All in all, we collected three snapshots of all market segments — one in May 2019, one in July 2020, and the last one in April 2021, respectively. With the data collected, we *characterise* the market as done in Section 3.3, then analyse skills statically (namely, *traceability analysis*) and dynamically (*interrogation analysis*) while performing a *differential analysis* to highlight changes over time.

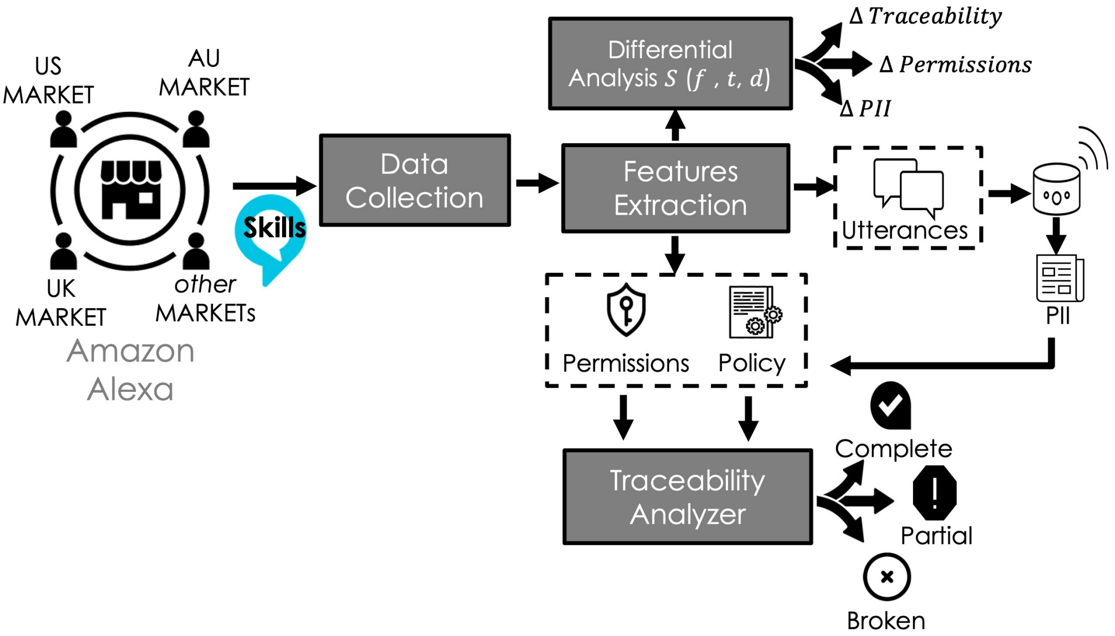


Fig. 5.1 Overview of the methodology

5.2.1 Traceability Analysis

We look at privacy policies to understand how developers disclose and justify the data permissions they request. For this, we leverage SkillVet. We focus only on the 5 English-speaking markets: US, UK, IN, AU and CA, which represent over 80% of the skills. Amazon lists these permissions in the market, and consent is given at installation time. The data actions defined in the privacy policies are extracted using Natural Language Processing (NLP).

5.2.2 Interrogation Analysis

We dynamically interact with the skills by systematically engaging in a synthetic conversation following the method in [88]. Our tool comprises a range of components designed to meaningfully interact with a skill (including utterance extraction, question understanding, answer generation and behaviour exploration) as described in detail in Section 5.7.1. Our tool has 81% coverage, similar to the coverage reported in [88].

5.2.3 Differential Analysis

We finally study how a skill changes by computing a differential of the representation of the skill at two points in time. Let the state of a skill be $S(f, t, d)$, where f is any of the features obtained during the *feature extraction* process (typically, the permissions, although our methodology supports a wide range of features), t is the result of the traceability analysis (typically, complete, partial or broken), and d is the result of the interrogation analysis (typically, data collection practices through conversations). We define the differential of two states as $\mathcal{D} = S_{t_1}(f, t, d) - S_{t_2}(f, t, d)$, where t_1 and t_2 are two points in time and $\mathcal{D}$ represents the Levenshtein distance between the set given as inputs. For instance, a skill i that requests a new permission p in 2021 (over 2020) and its traceability changes from complete to broken results in the following: $S^i_{2021} - S^i_{2020} = [\texttt{insert}(p), \texttt{substitute}(\textit{complete}, \textit{broken})]$

5.3 Skill Ecosystem

This section presents our characterisation of Alexa skills across 11 markets over the three snapshots in our dataset.

5.3.1 Skills and Developers

Table 5.1 shows the breakdown of the total number of Alexa skills across Amazon marketplaces. From the table, it can be seen that there are 124,026 skills published in 2021. This is 10.94% higher than the 111,796 skills published in 2020 and 46% higher than the 84,856 skills published in 2019. In addition, more skills were published between 2019 and 2020 (26,940 skills) than between 2020 and 2021 (12,230). Across all years, English-speaking marketplaces have the largest skills, representing over 80% of the skills. However, the Spanish market has the highest increment in the number of skills changing by almost 300% from 1,286 in 2019 to 5,435 in 2021. Likewise, we see more skills in the

Table 5.1 Number of Alexa skills from 2019 to 2021

Market	Skills					
	2021		2020		2019	
	N	%	N	%	N	%
US	68,667	31.83%	55,736	28.01%	51,338	30.82%
UK	37,056	17.18%	34,618	17.40%	29,094	17.46%
IN	28,672	13.29%	31,246	15.70%	20,989	12.60%
CA	27,093	12.56%	26,027	13.08%	24,700	14.83%
AU	24,512	11.36%	24,062	12.09%	23,123	13.88%
DE	10,631	4.93%	10,287	5.17%	8,928	5.36%
ES	5,435	2.52%	5,010	2.52%	1,286	0.77%
IT	4,649	2.16%	4,203	2.11%	2,210	1.33%
JP	3,637	1.69%	3,545	1.78%	2,679	1.61%
FR	2,863	1.33%	2,288	1.15%	1,341	0.80%
MX	2,486	1.15%	1,972	0.99%	897	0.54%
Total	215,701	100.00%	198,994	100.00%	166,585	100.00%
Unique	124,026	*57.5%*	111,796	*56.2%*	84, 856	50.93%

IN marketplace in 2020 (31,246) compared with 28,672 in 2021. Overall, there is a high percentage increase in the number of skills added to the non-English-speaking markets (71%) than the English-speaking markets (25%) over the years.

We also measure the number of developers operating in the ecosystem per marketplace. As shown in Table 5.2, there are 50,526 developers in 2021. This is an 8% increase from the 46,804 recorded in 2020. Overall, there is a 62% rise in the number of developers we see from 2019 to 2021. As we see with the total number of skills during the years, the Spanish marketplace also has the highest increment in developers changing by a similar percentage of 300%. From the table, it can be seen that the highest number of developers is also located within the English-speaking marketplaces.

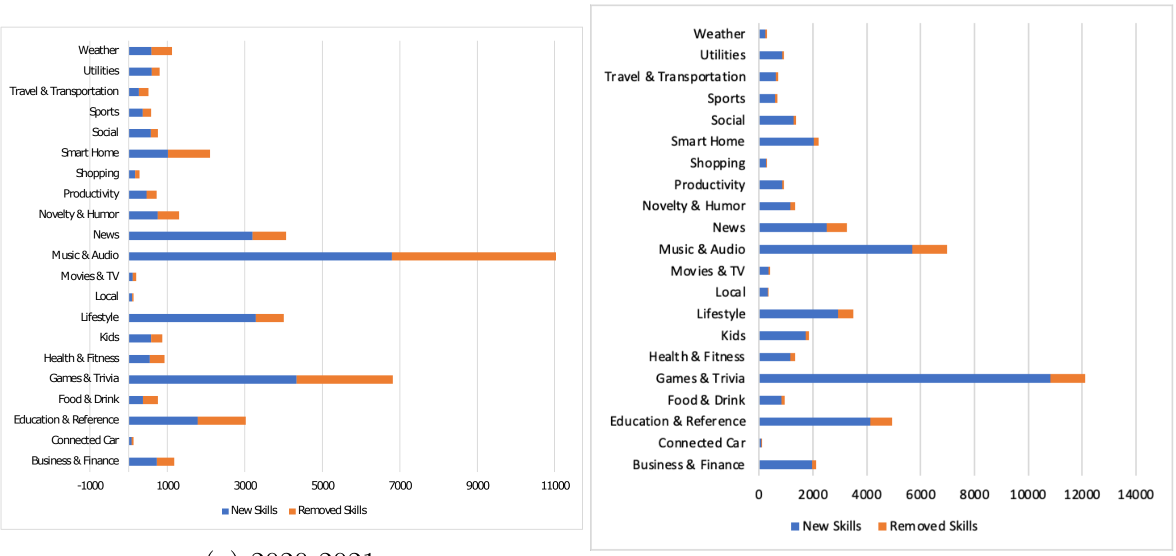
Table 5.2 Number of Alexa skill developers from 2019 to 2021

Market	Developers					
	2021		2020		2019	
	N	%	N	%	N	%
US	29,394	31.25%	25,483	28.88%	19,507	19.48%
UK	15,998	17.01%	15,066	17.08%	12,078	12.06%
IN	11,781	12.53%	13,316	15.09%	9,197	13.35%
CA	11,662	12.40%	11,509	13.05%	10,773	15.64%
AU	11,603	12.34%	10,762	12.20%	10,123	14.69%
DE	4,018	4.27%	3,713	4.21%	3,165	4.59%
ES	2,856	3.04%	2,543	2.88%	716	1.04%
IT	2,331	2.48%	2,049	2.32%	1,095	1.59%
JP	1,437	1.53%	1,377	1.56%	1,056	1.53%
FR	1,407	1.50%	1,194	1.35%	641	0.93%
MX	1,563	1.66%	1,212	1.37%	540	0.78%
Total	94,050	100.00%	88,224	100.00%	68,891	100.00%
Unique	50,526	*53.72*%	46,804	*53.05*%	31,238	*45.34*%

5.3.2 Skill Category

We look at the skill categories to understand how skills are grouped and study what changes have occurred over time, specifically by looking at the newly added and removed skills across the years.

Figure 5.2b shows by categories the number of skills that have been added to the ecosystem in 2020 with respect to 2019. Also, it shows the skills that have been removed in 2020 since 2019. The chart shows that the *Game & Trivia* is the category with the highest number of newly added skills. It contains about 27% of the total added skills. Similarly, Figure 5.2a shows by category the number of skills that have been added in 2021 from what we saw in 2020, as well as the skills that have been removed since 2020. The data shows *Music & Audio* as the top category with the highest number of newly added skills. It contains 6,796 (25.5%) new skills.



(a) 2020-2021

(b) 2019-2020

Fig. 5.2 Number of skills per category added and removed across the years

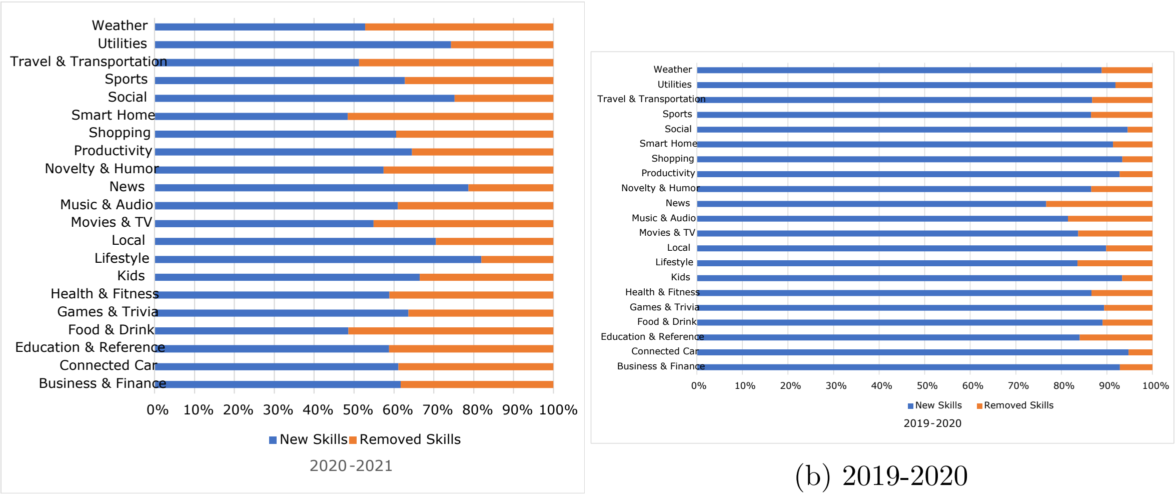
Looking at the removed skills over the years, from the data in Figure 5.2a, we see that the *Music & Audio* category has the highest number of removed skills across the marketplaces. In 2021, this category contained more than 50% of the total skills removed in the US market with respect to 2020. We also examine the interplay between skills added into a category and the number of removed skills across time. As shown in Figure 5.3a, we see that *Smart Home*, and *Food & Drink* categories in 2021 have more skills removed than the number of skills added. Overall, fewer skills are removed between 2019 and 2020 than the number of publications between 2020 and 2021.

5.4 Permissions

We next present a characterisation of Alexa skills through the lens of our dataset. In particular, we focus in this section on permissions as a reliable proxy to understand data collection practices [161, 204].

5.4.1 Distribution of Permissions by Skills

Table 5.3 shows that more than 97% of skills have not been requesting permissions over the years. However, most of the skills that request permissions appear listed in an English-speaking marketplace. Notably, the skills that declare permissions display an increasing trend over time. In particular, we see 0.41% of skills requesting more than



(a) 2020-2021

(b) 2019-2020

Fig. 5.3 Percentage of skills per category added and removed across the years

one permission in 2019, rising to 0.71% in 2020 and to 0.82% in 2021. We see similar trends as the number of permissions increases, e.g., there are 53 (62 – 9) skills more that are asking for >= 4 permissions in 2021 when compared to 2019. This increase is on average for all marketplaces, and we note that it is imbalanced. For instance, in 2021, the number of skills asking for more than four permissions in the IN marketplace increases by 70%, while the number of skills asking for three permissions increases by 133%.

Table 5.3 Number of permissions request over time

No	2021		2020		2019	
	Skills	%	Skills	%	Skills	%
0	120,848	97.44%	109,120	97.61%	83,427	98.32%
1	2172	1.75%	1882	1.68%	1082	1.28%
2	625	0.50%	511	0.46%	241	0.28%
3	239	0.19%	188	0.17%	83	0.10%
4	80	0.06%	57	0.05%	14	0.02%
>=4	62	0.05%	38	0.03%	9	0.01%
Total	124,026	100.00%	111,796	100.00%	84,856	100.00%

Table 5.4 Distribution of permissions per category

	Music	Games	Lifestyle	Education	Health	Productivity	Business	Shopping	Food	Travel	Social	Weather	News	Home	Utilities	Sports	car	Local	Kids
2021	459	319	293	247	223	220	210	175	163	153	152	98	93	92	67	60	37	36	32
2020	376	215	264	204	171 ▼	187	162	157	161	165	128	86	67	92	57	56	31	34	14 ▼
2019	122	157	119	117	231	73	105	52	74	80	58	48	23	42	32	16	10	15	24

5.4.2 Distribution of Permissions per Category

To better understand the relationship between skills in a category and the number of permissions requested, we selected and further analysed the top categories with many skills asking for more than two permissions. Our finding shows that out of the 21 skills requesting for more than three permissions under the *Education and Reference* category in 2021, 15 (71.4%) skills are developed by *VoiceXP* all asking for four permissions — *Mobile Number, Email Address, Full Name* and *Device Address*. Furthermore, 9 (60%) of these skills have no reviews or ratings. Likewise, in 2020, *VoiceXP* also has 12 (75%) of 15 skills with more than 3 permissions in the *Education and Reference* category. Similarly, in the *Music & Audio* category, 50% of the 20 skills requesting more than two permissions are also published by a single developer — *Alpha Voice*. These skills request *Device Address, Lists Read Access*, and *Lists Write Access* with 40% having a single review or rating.

5.4.3 Distribution of Permissions by Type

Table 5.5 shows how the different permissions are distributed across the years. The most requested permissions are *Device Address*, *Email Address* and *Device Country & Postal Code* generally used to offer services based on the user's location. For example, the *Device Address* is asked for by 772 skills (570 developers) in 2021, 753 skills (567 developers) in 2020, and 519 skills (381 developers) in 2019. In contrast, *Amazon Pay* is the least asked permission which is requested by 111 skills published by 75 developers in 2021 and 83 skills by 61 developers in 2020. Skills requesting for *Location Service* increase from 2% in 2019 to 4% in 2020 while those asking for *Device Address* and *List Access* reduces by about 9%, respectively. Overall, we see more skills asking for *Location Service*, *Email Address*, *Name*, *Reminders* and *Mobile Number* across the years.

Table 5.5 Distribution of permissions by type

Permission	2021			2020			2019		
	D	N	%	D	N	%	D	N	%
Device Address	570	772	16%	567	753	19%	381	519	28%
Email Address	445	761	16%	345	544	14%	137	160	9%
Device Country	394	707	15%	381	644	17%	305	378	20%
Name	287	524	11%	223	400	10%	79	118	6%
Reminders**	282	482	10%	205	263	7%	64	82	4%
Alexa Notifications**	275	555	12%	249	461	12%	117	165	9%
List Access	183	415	9%	183	417	11%	155	347	19%
Location Services	177	203	4%	140	156	4%	35	37	2%
Mobile Number	152	231	5%	112	162	4%	35	37	2%
Amazon Pay	75	111	2%	61	83	2%	31	31	2%
Timers**	15	15	0.3%	8	8	0.2%	-	-	-
Skill Personalization	5	6	0.1%	-	-	-	-	-	-
Total	2860	4782	100%	2474	3891	100%	1339	1874	100%
Unique	1887	3178	66%	1714	2676	69%	1022	1429	76%

D = Number of developers, **N** = Number of skills,
** Not considered by Amazon in the privacy requirements for skills.

On the contrary, fewer skills are now requesting for *List Access*, *Device Address*, and *Device Postal Code*. This could potentially be due to developers being increasingly more concrete on the type of personal information they collect.

Note that in Table 5.5, *Name* refers to the aggregate of the *First Name* and the *Full Name* permissions and *List Access* is the aggregate of *List Read Access* and *List Write Access* permissions. Also, *Alexa Notifications* permission is now deprecated.

5.5 Traceability

In Chapter 3, we looked at privacy policies to understand how developers disclose and justify the data permissions they request. We study the traceability between the data operations and the data actions defined in the privacy policies. Here, we look at the traceability of skills longitudinally to understand changes across time. For this, we use We focus only on the 5 English-speaking markets: US, UK, IN, AU and CA, as SkillVet currently supports only privacy policies written in the English language. Recall from Chapter 3 that English-speaking markets represent over 80% of the skills in this ecosystem.

5.5.1 Traceability per Skills and Developers

The chart in Figure 5.5 shows that developers' data disclosure practices were poor in 2020 compared to 2019. 35% of developers have skills with broken traceability compared to 51% in 2020. Instead, traceability improved considerably in 2021 compared to the previous years (see Section 5.6 to understand the factors impacting these changes, including the responsible disclosure of 675 skills we did in the second half of 2020). Similarly, in Figure 5.4, there are 11% more skills with complete traceability in 2021 than what exists in 2019 and 15% more skills when compared with 2020. On the other hand, the number of skills and developers with complete traceability decreases in 2020 from what we see in 2019. Note that the number of developers with sound data practices disclosure rose from 55% in 2019 to 57% in 2021.

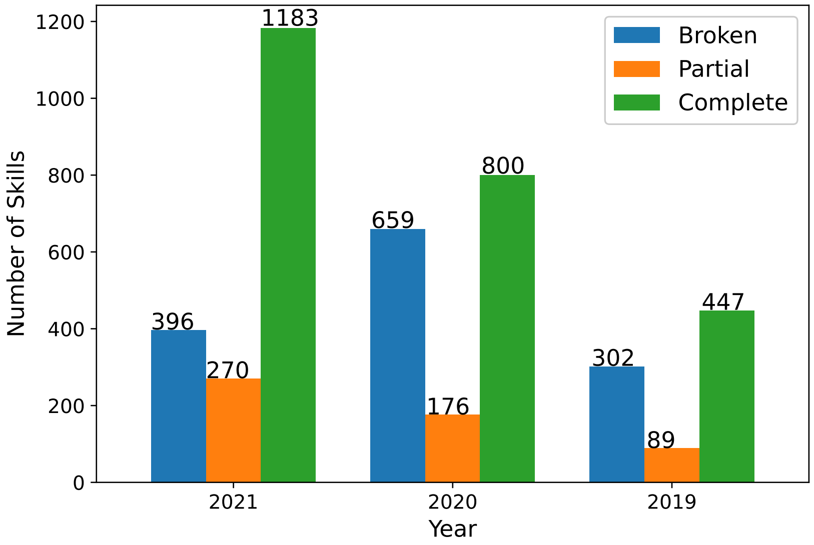


Fig. 5.4 Traceability results for English-speaking markets skills from 2019 to 2021

5.5.2 Traceability per Category

To understand how traceability changed across types of skills, we look at the market category. Specifically, we compute the traceability by category in the five English-speaking marketplaces. Next, we evaluate the different categories based on the number of concerns (broken and partial) normalised by the number of well-defined policies

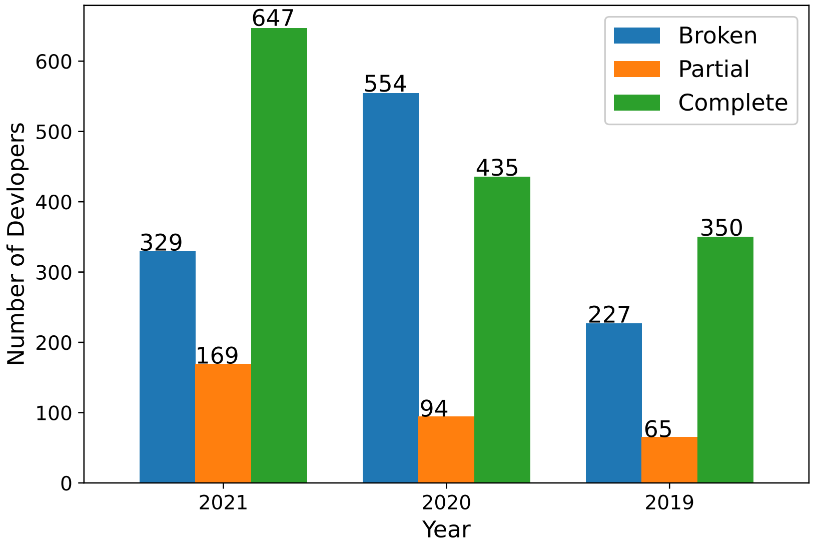


Fig. 5.5 Traceability results for English-speaking markets developers from 2019 to 2021

(complete). As shown in Table 5.6, the *Kids* category is ranked first in category with issues in 2021 and the *News* category in 2020. They have the highest ratio of skills with inadequate privacy disclosure to those that are well defined. The *Music & Audio* category has the largest number of complete traceability skills across the years, which is also a sizeable proportion of skills within the category.

Table 5.6 also shows that traceability improves in category such as *Business & Finance*, *Movies & TV*, and *Music & Audio*. For instance, the *Business & Finance* category is currently ranked 18th out of the 21 categories. This is an improvement from the previous rank of 12th in 2020 and 8th in 2019. We now see bad privacy practices in 51 skills compared to 152 skills in the same category with complete traceability. Also, the *Movies & TV* category ranked 10th in 2019 and 5th in 2020, now ranked 19th. We similarly observe categories where traceability has gone worse. An example is the *Utilities* category currently ranked 6th in 2021 from 9th in 2020 and 11th in 2019. Our findings here confirm the hypothesis drawn in Section 5.3 that certain categories are under heavier scrutiny, but it also shows that the effectiveness of having more complete traceability and less broken (or partial) in a category changes from one year to the another.

5.5.3 Traceability by Number of Permissions

To establish whether skills that request more permission are more traceable or not, we study the relationship between the number of permission requested by skills and their

Table 5.6 Traceability by category (markets in English)

Category	2021				2020				2019			
	R	B	P	C	R	B	P	C	R	B	P	C
Kids	**1**	2	4	-	19	2	-	4	6	10	2	9
Novelty & Humor	2	17	5	8	2	24	-	8	3	11	3	8
Weather	3	36	17	37	4	45	5	22	4	15	7	15
Food & Drink	4	53	25	61	3	71	20	40	13	19	5	27
News	5	12	34	36	**1**	27	24	11	7	10	2	10
Utilities	6	23	8	30	9	26	4	18	11	10	4	13
Games	7	66	**56**	151	8	121	**42**	101	5	66	**36**	75
Smart Home	8	24	8	41	7	44	2	27	20	16	-	27
Local	9	5	4	11	14	7	4	9	**1**	7	2	3
Connected Car	10	9	2	14	18	7	2	11	2	1	1	-
Social	11	17	5	30	16	22	3	25	17	9	-	12
Travel & Transp.	12	36	15	73	6	77	8	45	14	15	4	22
Health & Fitness	13	32	25	87	10	73	13	55	12	**131**	34	**170**
Shopping	14	13	43	91	15	20	**42**	59	9	12	3	13
Productivity	15	58	17	139	11	94	12	81	19	16	-	26
Lifestyle	16	**68**	35	198	17	122	12	147	16	36	11	65
Education & Ref.	17	61	21	201	13	**125**	7	113	15	42	9	67
Business & Finance	18	28	23	152	12	75	7	64	8	37	7	40
Movies & TV	19	3	-	8	5	8	5	5	10	2	-	1
Music & Audio	20	45	19	**367**	20	79	12	**292**	18	47	4	78
Sports	21	4	-	38	21	7	-	38	21	-	-	7
Total	-	612	366	1,773	-	1,076	224	1,175	-	512	134	688
Unique	-	396	270	1,183	-	659	176	800	-	302	89	447

B = Broken, **P** = Partial, **C** = Complete, **R** (Rank) $\sim$ (B+P)/(C+1)

traceability. The data in Table 5.7 shows that there is a higher number of skills with complete traceability, asking for just one permission.

5.5.4 Traceability by Type of Permissions

To understand how traceability varies across the different types of permissions over the years, we also look at the traceability of skills per permission requested. Table 5.8 shows the distribution of traceability across the different types of permission for the skills that request permissions and warrant a privacy policy in the English-speaking marketplace. The permissions are first grouped into broken, partial, complete, with respect to the policies of the skills where these permissions are requested. In 2021, 2,852 permissions are requested (622 by skills with broken traceability, 485 by skills

Table 5.7 Table showing the relationship between the number of permissions requests by skills and their traceability.

No of Permission	Traceability	2021		2020		2019	
		N	%	N	%	N	%
1	C	841	69%	590	51.7%	346	53.1%
	B	261	21%	474	41.5%	232	35.6%
	P	123	10%	77	6.7%	74	11.3%
	Total	1,225	100%	1,141	100.0%	652	100.0%
2	C	173	48%	103	35.6%	62	50.4%
	B	97	27%	129	44.6%	53	43.1%
	P	89	25%	57	19.7%	8	6.5%
	Total	359	100%	289	100.0%	123	100.0%
3	C	105	60%	68	48.2%	35	66.0%
	B	29	17%	40	28.4%	14	26.4%
	P	41	23%	33	23.4%	4	7.5%
	Total	175	100%	141	100.0%	53	100.0%
4	C	52	80%	35	64.8%	3	50.0%
	B	7	11%	14	25.9%	2	33.3%
	P	6	9%	5	9.3%	1	16.7%
	Total	65	100%	54	100.0%	6	100.0%
>=5	C	12	48%	4	40.0%	1	25.0%
	B	2	8%	2	20.0%	1	25.0%
	P	11	44%	4	40.0%	2	50.0%
	Total	25	100%	10	100.0%	4	100.0%

with partial traceability, and 1,509 by skills that exhibit complete traceability). We see that *Amazon Pay* is the least asked permission which is requested by 76 skills in 2021 and 51 skills in 2020, and also tends to be requested more by skills that have complete traceability. In contrast, *Location Services* permission requested by 137 skills in 2021, 94 in 2020, and 15 skills in 2019 is found more in skills that exhibit broken traceability. This means that the type of permission matters when it comes to the justification of the collection practices and the desired data flow patterns. This could be effectively leveraged to implement a better triage mechanism during a vetting process. We discuss the implications of over-privileged skills in Section 2.6.

Table 5.8 Distribution of traceability across different permissions in the 5 English-speaking marketplaces across 3 years

Permission	2021				2020				2019			
	R	B	P	C	R	B	P	C	R	B	P	C
Device Address	586	**162**	88	336	**559**	**255**	58	246	**374**	**141**	**40**	**193**
Device Country	**598**	87	48	**463**	528	148	25	**355**	273	87	20	166
Email Address	558	82	119	357	385	123	83	179	86	30	7	49
List Access	296	105	20	171	281	140	10	131	227	96	6	125
Name	419	64	**134**	221	322	93	**101**	128	87	25	30	32
Mobile Number	182	27	32	123	127	42	21	64	27	6	6	15
Location Services	137	41	45	51	94	46	21	27	15	7	4	4
Amazon Pay	76	7	16	53	51	9	11	31	13	1	3	9
Total	2,852	575	502	1,775	2,347	856	330	1,161	1,102	393	116	593
Unique	1,849	396	270	1,183	1,635	659	176	800	838	302	89	447

R = Requested, **B** = Broken, **P** = Partial, **C** = Complete.

5.5.5 Profiling Developers

Table 5.9 shows the number of developers per type of traceability considering the 5 English marketplaces across the years.

Complete: In 2021, there are 638 (56%) developers with *all* their skills showing complete traceability. This implies that all their skills have statements in their privacy policies clearly stating and justifying their request's permissions. This is higher than the 423 (40%) developers we see in 2020 and the 347 (54%) in 2019.

Broken: There are 540 developers in 2020 with *all* their skills broken. This accounts for about 51% of the developers. Their skills do not generally offer an adequate explanation when we analyse the skills, their privacy statements, and their reviews. The number is much lower in 2021 as we find only 323 (29%) developers with all their skills exhibiting broken traceability.

Partial: We see 161 developers with *all* their skills with partial traceability in 2021. This accounts for about 14% of the developers. They appear to have a lax attitude when writing privacy policies and informing users of how the personal information they request is used. We see the highest number of skills with partial traceability in 2021 compared to the 8% and 10% we see in 2020 and 2019, respectively.

Mixed: We see a handful of developers with a mix of broken (B), partial (P), and complete (C) (see B+P, etc. in Table 5.9). There is an interesting case of a developer *Blutag Inc.* in the P+B+C case. It has 74 skills, 1 broken (dead link), 35 partial, and 38 complete.

While we see an increasing trend towards having more complete traceability over time, we still see more broken skills in 2021 than in 2019. Also, partial traceability seems to be an issue as it continuously grows over time.

Table 5.9 Developers' disclosure practices

Year	D	B	P	C	B+P	B+C	P+C	P+B+C
2019	638	223	64	347	1	3	-	-
2020	1068	**540**	90	423	**3**	**11**	-	**1**
2021	1133	323	**161**	**638**	2	3	**5**	**1**
Total	2839	1086	315	1408	6	17	5	2
Unique	1349	666	182	740	4	13	5	1

D = Developer, **B** = Broken, **P** = Partial, **C** = Complete

5.6 Factors Impacting Traceability

Next, we explore several hypotheses on what could have influenced the changes we see over the years. In particular, we analyse: i) the impact of new skills on the ecosystem. ii) how existing skills' traceability has changed over time, iii) the impact of change in skills' permissions in the ecosystem, iv) the effect of the responsible disclosure we did to Amazon and third-party developers.

5.6.1 Effect of New Skills on Traceability

We investigate the effect of new skills on traceability. As shown in Figure 5.6 there are 996 new skills added between 2019 and 2020 that ask for permissions that warrant privacy policies. Similarly, there are 399 new skills added between 2020 and 2021 that ask for permissions that deserve privacy policies. Interestingly, this data shows that more skills with complete traceability have been added over the years than skills that exhibit broken or partial traceability. In particular, 518 (52%) skills in 2020 and 256 (64%) skills in 2021 are new skills added with complete traceability.

However, the number of skills with issues is also on the rise. In particular, 478 (48%) skills with privacy issues were added between 2019 and 2020, and 143 (36%) of these skills were added between 2020 and 2021. One good example is the “air monitor” skill by *AirMonitor* added in 2021. This skill collects *Device Address and Location Services*. However, the skill exhibit broken traceability as the privacy policy links direct users to a dead page. Although the overall state of affairs is improving, many newly submitted skills still have privacy issues. We, therefore, posit that the vetting process could still be improved. Also, the research community (with studies like ours, as we show in Section 5.6.4) has made a commendable effort to contribute to the market’s sanitation.

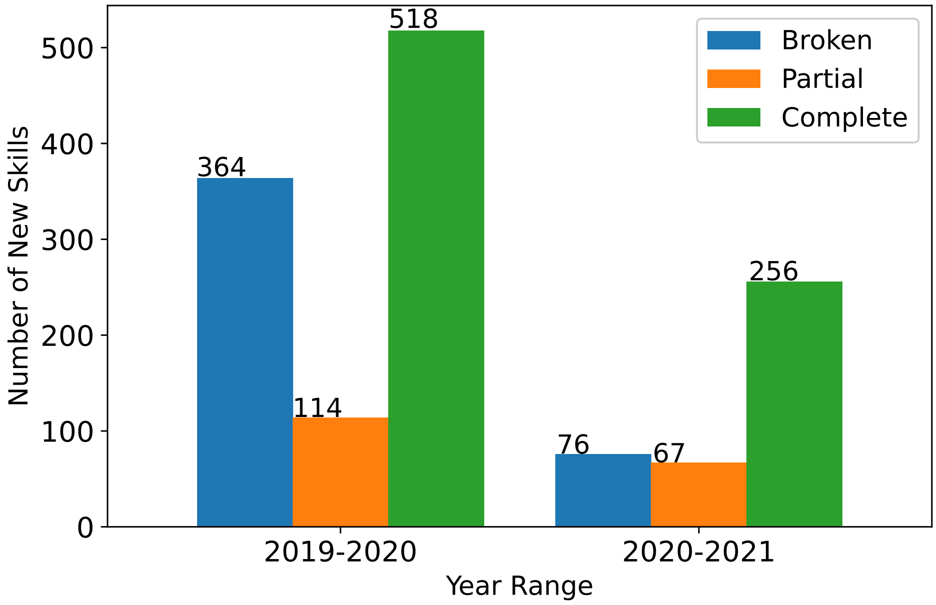


Fig. 5.6 Traceability of newly added skills

5.6.2 Traceability across Existing Skills

We investigate how the traceability of existing skills has changed over time. This could allow us to measure the effect of Amazon’s continuous vetting techniques. Table 5.10 shows how the traceability has changed over time. We could see that out of the 302 broken skills in 2019, 199 (65.9%) were still broken in 2020, 90 (29.8%) skills were removed, 2 (0.7%) have partial traceability, and 6 (2%) were complete in 2020. However, 80 skills that exhibit complete traceability in 2019 were broken in 2020. On further analysis, we found that this change in traceability is due to a lack of access to the skills’ privacy documents. The policy links are either dead or take users to a dead page. A possible explanation for this might be that developers no longer maintain these skills.

Table 5.10 Detailed change in traceability across the three years

	2020											
	Traceability	B		P		C		PR		SR		Total
	-	N	%	N	%	N	%	N	%	N	%	
2019	B	199	65.9%	2	0.7%	6	2.0%	5	1.7%	90	29.8%	302
	P	16	18.0%	58	65.2%	1	1.1%	1	1.1%	13	14.6%	89
	C	80	17.9%	2	0.4%	275	61.5%	3	0.7%	87	19.5%	447
	Total	295	35.2%	62	7.4%	282	33.7%	9	1.1%	190	22.7%	838
	2021											
	Traceability	B		P		C		PR		SR		Total
		N	%	N	%	N	%	N	%	N	%	
2020	B	315	47.8%	38	5.8%	162	24.6%	57	8.6%	87	13.2%	659
	P		0.0%	163	92.6%	2	1.1%		0.0%	11	6.3%	176
	C	5	0.6%	2	0.3%	763	95.4%	3	0.4%	27	3.4%	800
	Total	320	19.6%	203	12.4%	927	56.7%	60	3.7%	125	7.6%	1635

B = Broken, **P** = Partial, **C** = Complete, **SR** = Skills Removed, **PR** = Permission Removed

But, on the other hand, this could also be one of the reasons why skills remain broken over the years.

Equally, out of 659 broken skills in 2020, 162 (24.6%) were complete in 2021, 57 (8.6%) had their permission removed, 87 (13.2%) complete removed, and 315 (47.8%) were still broken. We see 5 (0.6%) skills that were previously complete in 2020 becoming broken in 2021 also due to dead link. An example is the "Kids Booklet" by *WebRecycles Inc* that collects *Device Country and Postal Code.* The traceability changed from complete in 2020 to broken in 2021. Even so, the traceability of skills with bad privacy practices in 2020 improved considerably in 2021. Only 320 skills were still broken from the same set of skills we see in 2020 compared to 659. The result shows that both Amazon and developers have worked to improve the traceability of skills in this ecosystem.

5.6.3 Effect of Change in Permission(s) on Traceability

Does traceability change because permissions change? To answer this question, we first study changes in the use of permissions. In Figure 5.7, we see an increase in the number of permissions per skill and how it negatively impacts their traceability. The "PRE" and "POST" suffix in Figure 5.7 indicates pre-increase and post-increase, respectively. Between 2019 and 2020, 14 skills asked for additional permissions. In

2019, 11 (79%) skills had complete traceability, while 3 (21%) have inadequate privacy disclosure practices, including partial traceability. However, the number of skills with privacy issues increases by 100% to 6 in 2020 after the skills requested more permissions. A similar trend can be seen between 2020 and 2021, where the number of skills with insufficient privacy disclosure increases by 100%. For example, "Salah Time" skill by *Arshad* collects *Device Country and Postcode* in 2020 and exhibits complete traceability. It then collects *Device Address, Location Services, Reminders* in 2021 and exhibits partial traceability. The traceability difference from complete to partial is due to the use of the same privacy policy, even when different data is collected.

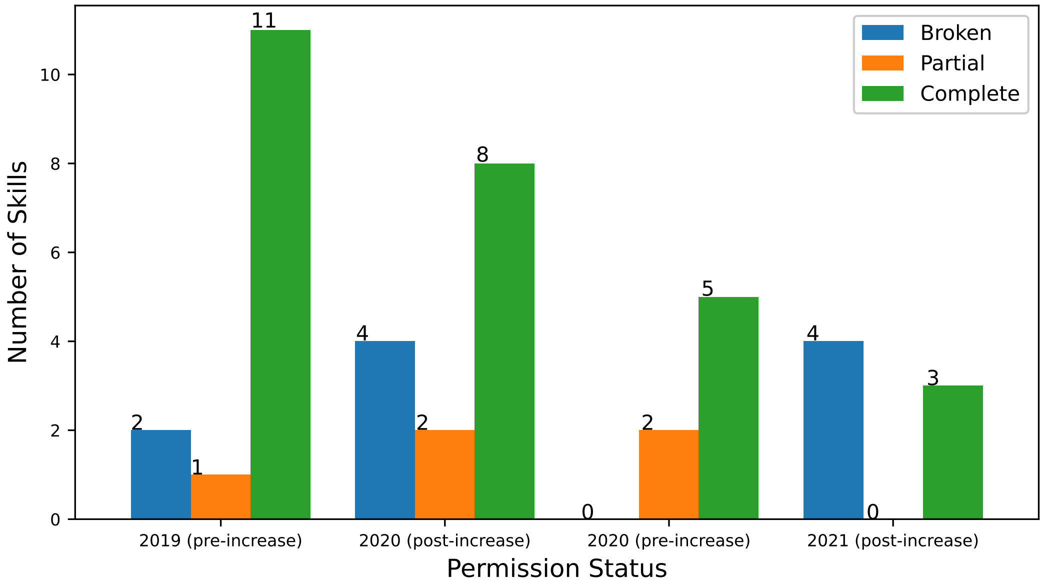


Fig. 5.7 Traceability of skills before and after the increase in the number of requested permissions

We next look at the opposite angle and study changes in traceability as the number of permissions decreases. Figure 5.8 shows the traceability of skills before (PRE), and after (POST), they reduce the number of permissions they requested. Note that we exclude those skills that have their permission wholly removed to avoid biasing the result. As we can see, there is no change in the number of skills with privacy issues between 2019 and 2020, even after the number of permissions requested reduces. However, we can see an improvement in traceability when the number of permissions requested by skills reduces between 2020 and 2021. We see a 50% increase in the number of skills with complete traceability from 4 to 6 skills. Nevertheless, these results need to be interpreted with caution because of the small skills involved. However, the low number of skills does not mean a low interactions (or "installations"). For example, among the skills is the popular "Uber" skill by *Uber.com* with hundreds of reviews and possibly hundreds of interactions.

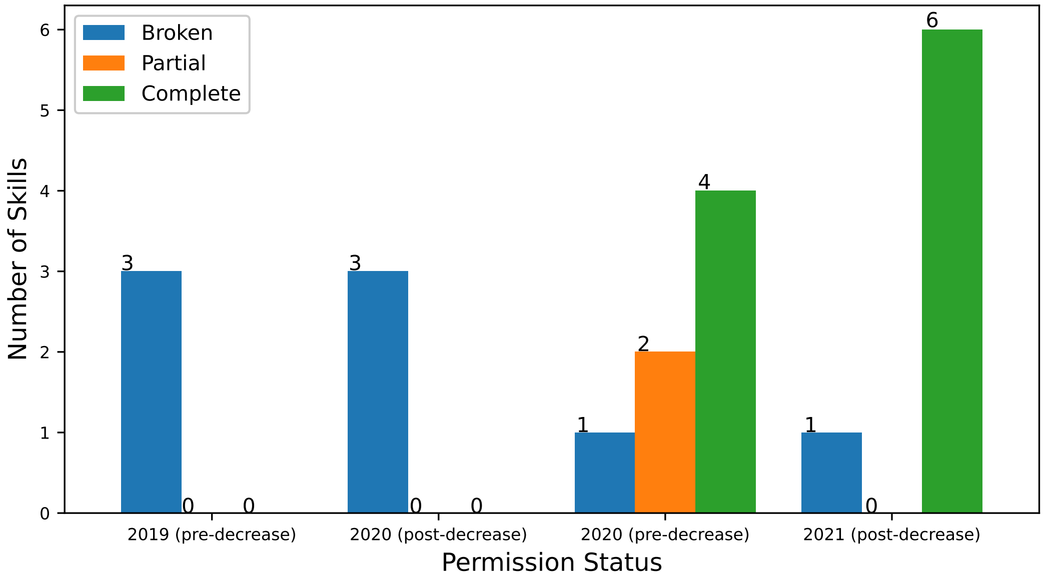


Fig. 5.8 Traceability of skills before and after they reduce the number of requested permissions

5.6.4 Effect of Responsible Disclosure

As mentioned in Section 3.6.4, we perform a responsible disclosure process, starting from mid-August 2020, reporting 675 skills with privacy issues to Amazon and the affected developers. Thus, we measure the effect of the responsible disclosure we did to Amazon and developers of skills with issues about their data disclosure practices.

From the data in Figure 5.9, we can see that out of 246 skills with broken traceability reported (*BROKEN PRE*), 111 (45.12%) no longer pose a threat to users at the time of writing: 45 (18.29%) of these skills have been removed and are no longer available on Alexa, 24 (9.76%) have their permission(s) removed, and 41 (16.67%) of them now have complete traceability. Overall, 356 (52.74%) out of 675 reported skills no longer threaten the users. This result corroborates our earlier findings in Section 5.6.1 that while traceability has improved, there are still skills with privacy issues across markets. Likewise, it shows how Amazon could benefit from enabling more actionable research mechanisms to study privacy issues.

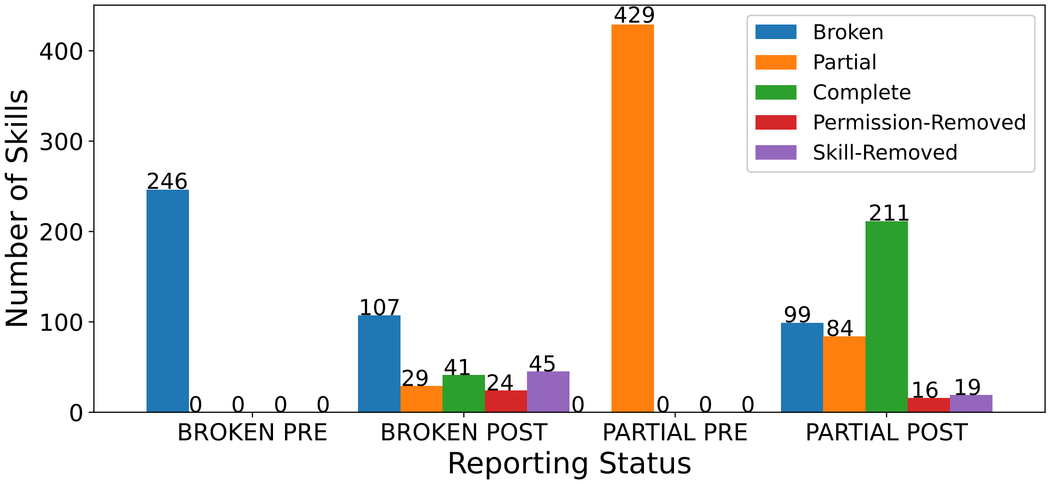


Fig. 5.9 Traceability after reporting the skills with issues to Amazon and developers.

5.7 Beyond API Permissions

As explained before, Amazon enforces access to personal data through a permission model embedded in their APIs. However, prior work [88] shows that skills could bypass this system and request personal data directly from the user via conversations. In particular, they found 100 skills across the Alexa US market in 2020 asking for personal information via conversation using an interactive system called SkillExplorer. To understand how conversational skills may have changed over the years, we study those available in the US market at the time we conduct our experiments (i.e., in 2021) and compare our findings with the results obtained in [88], which date back to 2020.

5.7.1 Tool Implementation and Evaluation

We implement the dynamic interactive tool SkillExplorer reported in [88] to interact with skills automatically. Note that the original implementation of SkillExplorer is not publicly available, nor it was given to us upon request. As in [88], our implementation follows a black-box approach to interact with skills, since the skills' code or executable is not available — recall that skills *run in the cloud instead of the users' device*, e.g., as an AWS Lambda function [18] or in a server controlled by the skill developer [19], and the only way to interact with them online is via a conversation. Our tool comprises four key components. i) The utterance extraction, where it extracts utterances from the skill page to initiate the conversation, ii) the question understanding section, to understand the response from the SPA, iii) the answer generation unit that generates a suitable answer to the question extracted from the SPA response for further interaction, and lastly iv) the behaviour exploration component that ensures that all routes of conversation are explored.

5.7.1.1 The Utterance Extraction

We extract the sample invocation utterances from the skill introduction page to activate the skill and initiate the interaction with Alexa. Developers are requested to provide sample utterances questions to help the user understand how to use the skill. These can be located by looking at the “a2s-utterance-box-inner” tag in the source code of the skill web page on the Alexa store.

5.7.1.2 The Question Understanding

After the first extracted utterance is sent to a skill, Alexa responds with the feedback and output from the skill. The feedback could be an answer to a request or a request for further commands. Our tool is implemented in a way that it could adequately understand the type of feedback given by Alexa. To understand the response from Alexa, we use Standard CoreNLP parser [213] to process the response as it considers clause level, phrase level, and word level when generating the abstract syntax tree from a text. This allows detecting patterns within the text at a lower level which can help identify and categorise specific questions. We consider five different types of questions:

1. Wh-Questions – These types of questions are open questions that users answer based on their understanding. An example of this question is, “Tell me your first name?”.

2. Yes/No questions – these are questions that expect “yes” or “no” answers. Examples include questions such as “Did you mean Lite Rock 105?”, “Do you want to listen to another fact?” that could be answered by responding with either “yes” or “no”.

3. Instruction Questions – this type of question contains instructions on how to answer them. It commonly includes the word “say” or “ask”. An example of this question type is “Please say repeat to hear the question again”, where the user is instructed to say “repeat”.

4. Selection Questions – this type of question gives users options from where they can select. An example is “To get started; you can get a quote or listen to the

daily briefing". Here, the user has two options to select from when generating their response.

5. Mix questions – this type of question contains more than one type of the other question type. For example, the question "You have started Crypto Ticker. Please ask me for a cryptocurrency price by saying, what is the price of bitcoin? Or, tell me the price of Ethereum" comprises Wh-question, instruction question, and selection question.

5.7.1.3 The Answer Generation

After categorising the questions, we generate a suitable answer for the question type. The answer to be developed need to keep the conversation going as much as possible. We can directly extract the answer from the questions themselves for the instruction, selection, and Yes/No questions. However, for the wh-question, we create a knowledge database to answer the question and explore the skill behaviour. We likewise leverage kuki[1] chat-bot due to its performance to answer other wh-question that are not covered in the knowledge database. Regarding the Mix question where multiple questions were detected, we prioritise answers as follows. If selection question and instruction exist simultaneously, we process both questions; if the Yes/No question exists, we answer with "yes" or "no".

5.7.1.4 The Behaviour Exploration

For a specific Alexa response, there could be multiple answers. To ensure that all conversation flow routes are explored before moving on to the following utterance, we use a tree data structure to represent the exploration status and track which question has been visited. Each node of the tree is a single interaction that comprises an Alexa response and the generated answer. When the tool interacts with the skill, the tree is drawn simultaneously. Thus, the tool ensures that every execution path is explored and all nodes are visited.

We use the Alexa simulator in the developer console for the interaction. The simulator allows developers to test their skills as they can directly feed text input into a skill and observe its outputs. For a more detailed explanation of the interactive system, we refer the readers to [88]. An important observation is that while the interactive tool automatically enabled the skill on the Alexa store before invocation, Alexa still has

[1] https://chat.kuki.ai/chat

issues understanding some of the skill invocation sample utterances. For instance, when we invoke the skill "Little Figure Skater Test" by *Modal Systems Ltd* with the sample utterance "Alexa, Start little figure skater test", Alexa responded with "Hmm, I don't know that one".

5.7.1.5 Evaluation

To check for the accuracy of our implementation, we conduct the same evaluation reported in [88]. In particular, we randomly selected 50 skills from different categories and manually interacted with them. The interaction generated 61 Mix questions, 14 Wh questions, 18 Yes/No questions, 11 Selection questions and 15 instruction questions and lasted for 5 hours. We then compared the output from the manual interaction with that generated by the interactive tool. The tool generates 97 outputs, which is 22 outputs less than the outputs from the manual interaction. The coverage implies that our tool has 81% coverage, similar to the coverage reported in [88]. Regarding the answer generation accuracy, all the Yes/No answers are correctly identified, and 9% of Mix questions were wrongly identified. On average, only 7% of answers are wrong.

5.7.2 Results

We interact with 35k skills in the US market, excluding skills without unique invocation names, as SkillExplorer can not handle them [88]. We find 65 skills requesting personal information via conversation. This is 35% less than the ones found in 2020 [88]. In particular, 58 (85%) skills collect users' name, 4 skills collect zip code, 3 (5%) request for user's birthday, 2 (3%) collects user's phone number, and a skill collects user's location.[2] We then use SkillVet to examine the traceability exhibited by these 65 skills found in 2021. The results show that 37 (57%) of these skills have *broken traceability* between the data collected via conversation together with the Amazon Alexa API and the data practices mentioned in their privacy policies, if any. Furthermore, 3 (5%) exhibit *partial traceability* and only 25 (38%) have *complete traceability*. Out of the 37 skills with broken traceability, we see that 29 (78%) skills completely lack a privacy policy document, five skills have a policy link that redirects us to a dead page, and three skills do not mention any data practices in their privacy policy document. Interestingly, most of the conversational skills we see requesting personal information via conversations do not ask for permission via Alexa API and only do it through conversation.

[2] 3 of the skills requests for more than one personal data.

One example is “F1 forecast” by *Jordan Perkins* in the US market, which informs its users of the latest news and updates in the F1 world. When the skill is invoked with the utterance “Alexa, ask f one forecast where Charles Leclerc qualified at the last race”, it requests to know the user’s address. The user’s address does not seem relevant to answer such a trivial question. The skill also lacks a privacy policy to state the purpose behind the data collection. Similarly, we see “Pick a book” by *Ju’s Apps*. This skill is in the US market, and according to its description, it recommends books to the users. However, when the skill was invoked with the utterance “Alexa, open pick a book”, it responded with “Hello and welcome to Pick a book. Not sure what to read? Then let me select a book for you. First of all, what is your name?”, but the skill does not have a privacy policy to justify this collection and state how the data will be stored and processed.

We also find skills such as the “Name Expansion” by *Jackson Jacob* asking for the name of the user to perform its function. Since the skill function expands the user’s name, we deem this relevant as the skill needs it to offer its services. However, while the skill has a privacy policy link, accessing it takes us to a dead page. Thus, the skill also exhibits broken traceability as it lacks a privacy policy document to justify its data practices.

5.7.3 What Has Changed over Time?

The above measurement shows a unique view of the underlying issues behind conversational skills requesting personal information. However, to understand changes over time, we further explore the 100 skills (developed by 89 distinct developers) collecting personal data via conversation provided to us by the authors of SkillExplorer [88]. This dataset was collected from the US marketplace in early 2020.

Out of the 100 skills reported to have privacy issues in 2020, we only find 25 conversational skills available in 2021 (which are also included in the 65 skills our tool finds, as stated above). Interestingly, from the 100 reported skills in 2020, only 3 skills have been taken down, and 72 are still listed in the Amazon market but unavailable. Amazon does not allow users to interact with those skills for several reasons. First, Alexa systematically suggests a different skill (albeit with a similar name) to the one being invoked. We see this in 36 of the 72 skills. Second, Alexa replies that it does not ‘understand what you want’ for 34 of the 72 skills. Note that we invoke skills through the Amazon Alexa simulator that supports text interaction using a simple command:

"Alexa open *[Skill Name]*". For another skill, Alexa replies that it is 'having trouble accessing' it. Finally, the remaining skill is listed as 'not currently available'.

Out of the 25 skills that are available in both 2020 and 2021, 15 skills (60%) still request users to provide personal information via conversation. Furthermore, only 5 (33%) skills exhibit complete traceability with the personal data collected via conversation. One has partial traceability, and the rest 9 (60%) have broken traceability. In particular, 7 have no privacy policies, one has a dead link and one has a privacy link that points to a porn site. This is the case of the "Praise Me" skill published by *Jackson Jacob*, which is available across all five English-speaking countries.

With most of the available skills still having broken traceability and no privacy policies, our takeaway is that, even if the number of skills collecting personal information via conversation is not that high (especially when compared with skills collecting information via permissions), more efforts are still needed to sanitise the ecosystem by Amazon.

5.8 Beyond Complete Traceability

To see whether skills could still be harmful even when exhibiting complete traceability, we look at the skills requesting permissions and assess if the justifications provided are convincing. To do this, we randomly select 100 skills from the set of 1,183 with complete traceability in 2021 we identify in this study. We look into the skills description pages and their privacy policies to understand if they clearly state why the permissions are requested. Additionally, we manually install them and interact with them.

We did not find any good reasons for some of the requested permissions in 7% of the skills from the interactions and examination. One good example is the "Artificial Intelligence (AI) Facts" skill developed by *by rbashish* in the UK market. According to the description page, the skill tells users about facts and figures for Artificial Intelligence. The skill requests access to the user address with the pretext to offer a better service. While the skill exhibits complete traceability (it acknowledges collecting the data), we deem this collection not relevant for a simple reason: the skill only answers trivial contextless questions. We see the same answers regardless of the location given. Note that the skill is a single interaction skill that terminates its interaction after performing one task.

Another example is the "Cork's 96 FM" published by *Wireless* in the UK market. The skill asks for: *Device Address, Full Name, and Email Address* permission. The developer acknowledges collecting the permissions and states this information is used for legitimate interest, fulfils a legal obligation, and personalises users' experience, which means the traceability is complete. However, our analysis shows that these permissions are irrelevant to their functionality. The skill lets users listen live to Cork's 96 FM and do not need to know the user's name, mobile number, or email address to offer this service.

These findings show that even while skills adequately discloses their data practices, there is evidence of skills being over-privileged. This means that using tools like SkillVet may not be enough. Therefore the research community and Amazon vetting process require a more sophisticated mechanism to account for this threat model.

5.9 Discussion

In this section, we present the essential findings of our work. We discuss what has improved and highlight areas that need further attention.

5.9.1 Increase in the Number of Skills and Developers

The result of this study indicates an increase in the number of third-party developers accessing the skill ecosystem. In Table 5.2, we see the number of developers rose by 62% from 31,238 in 2019 to 50,526 in 2021. The number of skills is also growing with the most growth (300%) observe in the Spanish market across the years. This number is a sizeable set and indicates that the ecosystem is ramping up. Unfortunately, while the growth offers users more functionality, it could potentially usher in a new level of threats and threat actors that could attack it. This is the most important reason to protect the ecosystem. After all, developers have different motivations for publishing skills (see Section 3.6.2) and being able to identify malicious developers could be vital in securing users' privacy. Thus, Amazon should implement a mechanism to validate the skill developer's identity for easy attribution, which is currently not possible, at least from the marketplace.

5.9.2 Improved Skill Review and Certification

Amazon's privacy requirements for skill developers mandate that a skill must come with an adequate privacy policy if it collects personal information [15]. But, unfortunately, we see skills having a privacy policy only to fulfil Amazon requirements and not to create awareness of data practices and privacy control which are essential to help the user protect their privacy. One good example is the "clean air check" skill published by *Laurence*. This skill collects *Device Country and Postcode* via Alexa API from the user. However, the content of the skill privacy document reads, "This skill knows nothing about you and stores nothing." The skill not only has broken traceability as it provides no data implication in its privacy policy document, but the document's content contravenes the skill data practices. This finding goes back to the lack of adequate privacy review when vetting a skill security profile that requires a privacy policy. It is apparent that developers are approaching privacy policy requirements as a tick box exercise disregarding the user's privacy. Ensuring that skills privacy policies are relevant, accessible, and understandable will go a long way in providing transparency about skill data practices and allow users to exercise the available privacy settings.

Overall, there seems to be an improvement in the review conducted as part of the skill certification process judging by the improved traceability of new skills added recently, as shown in Figure 5.6. Also, we have seen Amazon taking action by removing several skills with privacy issues that were reported to them. Even as many skills are requesting more permissions in 2021 (c.f Table 5.4), there is an improvement in skills traceability over the years. Notwithstanding, many newly added skills still exhibit broken traceability between the data operations evident to the users and the data actions defined by developers in the skill policies, which suggests there is still room to improve the review and automated tools such as the ones used in this study would help in that endeavour.

5.9.3 Better but Still Not Good Enough

In 2021, we see bad privacy practices in about 666 skills (36% of those that request permissions in the English-speaking marketplaces). This is an improvement from the 835 skills (51%) we observe in 2020 and also, in proportion, from the 391 (47%) we see in 2019. We also see how the research community has supported the sanitation of the market. All this seems to suggest an improving trend in terms of traceability, despite the high number of skills still exhibiting bad privacy practices in 2021. Notably, we see that 107 out of 246 skills (43.5%) continue to display broken traceability almost one year after being reported to both Amazon and the respective developers as part of our work (c.f. Section 5.6.4). We see that including or removing new permissions has a clear impact on traceability, with skills increasing the number of permissions across the years negatively impacting their traceability and skills decreasing the number of permissions impacting their traceability positively.

Furthermore, looking at the privacy issues based on skill categories, we see a large number of skills in the *Lifestyle* and *Games* category exhibiting broken traceability and partial traceability, respectively. These two categories comprise skills that offer services related to the user's behavioural pattern, daily interaction, consumption, work, activity and other interests that could potentially describe them. In contrast, the *Music & Audio* subcategory has the most significant number of complete traceability skills, which is also a sizeable proportion of skills within the subcategory. Note that this category is related to industries with a larger tradition of offering services on the Web, where privacy has been under scrutiny for longer.

5.9.4 Permissions vs Conversation

At the moment, data collection via conversation does not offer the same level of transparency compared to data collected via the Alexa API. This is mostly because data collected through the Alexa API is enforced by permissions, and this way, users can easily withdraw their consent. In fact, some skills direct users to visit Alexa companion apps to grant them access to the personal data they need. A good example is the "Barkibu" by *Barkibu*, which says "In order to send you an email report at the end of the consultation process, Barkibu will need access to your email address. Barkibu will also need access to your name to refer to you in a more personal way. Visit home screen in your Alexa app and grant me permissions."

We note that the vast majority of skills collecting data via conversation lack a privacy policy. Instead, those collecting via Alexa API permissions have a much higher proportion of complete traceability. This may suggest that there is stricter scrutiny when developers use the API to collect data. However, the API currently supports a limited number of permissions, and developers require to use alternative methods to collect information like *age, gender, relationship status*. Also, there may be questions about the impact on the user experience when forced to use other modalities rather than voice and/or the usability of such controls.

5.9.5 Unconvincing Justifications and Control over Flows

We identify over-privileged skills in about 7% of the skills exhibiting complete traceability. While these skills state the data they collect and justify their use, this justification may not be fully convincing. This implies that the research community, especially Amazon, needs to look beyond traceability and consider data relevance in the skill review and certification process. There is reasonably ambiguity about how data is used in practice by third-parties and the risks that data-starving apps pose to users. The Cambridge Analytica scandal [52] is an excellent example of these risks and why collection practices need to go under tight scrutiny.

There is a need for future research to study how to implement a framework that allows users to pronounce their intended data flow patterns. Similar frameworks [73, 104] have been successfully applied in smartphones for IoT apps, and recent work has studied users' desired data flow in SPA [2]. In our work, we go beyond understanding traceability issues as an important step forward towards understanding the privacy implications behind the use of third-party skills in SPA.

5.9.6 Limitations

Although the study has successfully highlighted how the Alexa skill ecosystem has evolved over three years, it has certain limitations. One important limitation is that we only conduct the traceability analysis for English-speaking marketplaces, as this is the language supported by SkillVet. However, the English market represents over 80% of the skills; hence, the result of the findings is a good representative of the current state of affairs in this ecosystem.

In addition, the automated analysis tools we use rely on NLP and ML and thus, inherit their limitations. Nevertheless, we believe that the (93%) accuracy achieved by SkillVet and 81% coverage level of SkillExplorer [88] is a good starting point for a meaningful analysis. Furthermore, SkillExplorer only works well with skills that have unique invocation utterances. It is challenging to explicitly specify which skill to invoke when many skills use the same invocation utterance. While we ensure that the skill of interest is enabled before invoking it, this workaround does not always work as Alexa will only invoke one of the skills based on its predefined algorithm. This implies that some skills will still not be activated even for the best effort.

5.10 Conclusion

The present study was designed to measure the Amazon Alexa privacy practices across three years, highlighting how the ecosystem has evolved and identifying the key factors that influence these changes. In particular, we examined the developers' data disclosure practices and presented the landscape in the third-party ecosystem. While the overall ecosystem has improved, newly submitted skills still pose an important risk to users' privacy. The vetting in the Amazon marketplace appears to suffer from important flaws, although the research community has made a commendable effort to improve the market's sanitation. Amazon would benefit from enabling more actionable mechanisms for researchers to study and analyse privacy and security issues in Alexa.

This chapter has shown how SkillVet could be leveraged to perform the traceability of skills at scale and help identify skills with bad privacy practices. However, it is vital to implement a traceability tool for others to use regardless of their technical expertise to help raise privacy awareness. In the next chapter, we present our implementation of SkillVet as an online Voice Assistant Privacy Assessment tool (VAPA).

Chapter 6

Voice Assistant Privacy Assessment Tool (VAPA)

This chapter addresses the final objective of this thesis, OBJ4: *To design an open tool that enables proactive audit of SPA skills privacy practices.* We extend the work in Chapter 4 by implementing a web application privacy tool based on SkillVet that helps in the process of identifying potential privacy issues in third-party voice-driven applications. This work was funded by the Information Commissioner's Office (ICO) as part of the project that aims to understand third-party developers' data disclosure practices in the Alexa ecosystem. The resulting web application is currently hosted on the KCL domain at https://skillvet.nms.kcl.ac.uk/.

6.1 Introduction

Privacy is a crucial factor in trust relationships. When users disclose their data to an organisation/third-party, they have to implicitly or explicitly trust the organisation/third-party with their data [101]. In SPA, privacy can be observed as trusting SPA providers, developers, and third-parties to properly handle the collected and generated data. These trusted entities are expected to disclose their practices via privacy policies.

However, as we highlighted in Section 5.9.2, they often fail to empower users towards making informed decisions. Moreover, these policies are sometimes challenging to access and poorly written, resulting in fewer users reading them. Failure to reading privacy policies may lead to using privacy-sensitive applications without awareness of the data practices involved [252, 136]. Since data collection, retention, and sharing are necessary for these devices to operate effectively, failure to read privacy policies may expose users to privacy risks [110]. Unfortunately, even when users read these privacy policies, they are often lengthy, generic and nonexclusive [179], making it hard to comprehend and time-consuming, and thereby fail to help improve users' awareness [207]. Hence, it is desirable to help users understand the data practices in privacy policies to offer transparency and help them know what choices are available to them and how to exercise them.

This chapter presents our implementation of SkillVet as an online Voice Assistant Privacy Assessment tool (VAPA) designed to help simplify users' understanding of data disclosure practices in skills' privacy policies. VAPA provide an interactive interface that could help users identify potential privacy issues in skills regardless of their technical background.This could help users know what options they have regarding their data and understand the implications of their choices. It could likewise help developers write better privacy policy documents with relevant data practices and assist regulators in detecting voice applications that violate existing regulations.

6.1.1 Permissions in SPA

Permissions are requested to enhance the users' conversations with content from external services, offer customised content, or perform specific tasks (Section 3.4.1). Before the skills are distributed to the public, they are subjected to a vetting process. Here, the SPA providers extensively review them to ensure that they only request the most definitive set of permissions required to perform their tasks. However, as we see in Section 2.4, malicious developers can bypass this process [44]. Besides, the platform providers don't have access to the skill source code during vetting; hence, there are limits to what measure they could implement.

In addition, hoping that a user will make an informed decision by notifying them about the permissions a skill requires before installing might be ineffective [199]. When enabling a skill, the user is shown a screen that details what information the skill intends to access. A user must explicitly grant this access to continue with the installation. However, most skills need to access multiple permissions. When a user sees almost the same notification for virtually every skill, they are likely not to pay attention to these prompts [70]. Hence, users may not be consistently aware of what the skill can do/is doing.

6.1.2 User Interfaces for Privacy Policies

Numerous strategies have been proposed to make privacy policies more accessible to users. Polisis [92] for instance, retrieves and presents policy paragraphs relevant to a user's question in a chatbot. Other works have studied the evaluation and presentation of privacy policies. For example, a study in [112] presents lengthy privacy policies in a nutrition-label-like form. Kay et al. [111] show that the visual elements, such as factoids, vignettes, iconic symbols and typography, increase the attention and retention of the users when reading the software agreements. Other research [124, 229] uses a comic-based interface to draw users' attention to privacy notices and terms of service agreements.

6.2 System Requirement and Analysis

The proposed VAPA application must satisfy several requirements to fulfil its objectives and make it appealing to users. In this section, we discuss the application's technical requirements and features. The application must have the following features:

6.2.1 Technical Requirement

1. Responsive design: There should not be any restrictions to accessing the web application's information. The user interface must be responsive to different screen sizes, such as desktop screens, tablets or mobile phones and must be compatible with multiple browsers.

2. Coherent visual design: The web contents and features should foster positive emotions toward traceability analysis, the application's primary function. Therefore, the application must have a simple layout with explanatory text and links that clarify how to use the web application.

3. Intuitive navigation: The ability to find information is also as important as the information itself. Users must be able to find their intended information with ease. The featured information must be clear, easy to understand, and the web application has intuitive navigation and workflows.

4. Fast performance: It needs to support multiple concurrent users and afford a fast online rendering of large datasets regardless of user network bandwidth for a better user experience.

5. Code reusability and maintainability: The code should be readable, easy to maintain and extendable.

6. Handling exceptions: The application should satisfactorily handle errors and inform users about its state while running.

6.2.2 VAPA Feature Requirement

1. Traceability Checker: The system needs to provide data querying to allow users to analyse skill traceability. In addition, users should have a choice in the manner they want to perform the traceability check.

2. Traceability Result Display: Users should be notified of the traceability result, and the system should display the privacy statement that contains data implications if any.

3. Interactive Statistical Display: Users should be able to explore general statistics about the skill ecosystem interactively.

4. Longitudinal Traceability Result: Users should be able to explore traceability results of skills longitudinally across years

We next describe how the proposed system architecture satisfies the requirements mentioned above.

6.3 System Architecture and Design

As shown in Figure 6.1, VAPA utilises a client-server framework that divides tasks between servers and clients. A server component provides services to clients and waits for them to be requested. A client component asks the server for a service, which the server either rejects or fulfils and returns to the client [26].

The user directly interfaces with the client-side, and the server-side supports the client-side and performs the detailed data processing. This framework permits centralised system management, allowing all necessary data and applications to be hosted in a single place. Besides, all nodes in the client-server system are independent, enabling easy upgrades, replacements, and relocation of the nodes. To implement the requirements listed in Section 6.2, modern web development techniques and tools will be used. The front-end part of the application relies on Html, CSS, and JavaScript frameworks. The backend is needed to support the essential features outlined in the requirements. We leverage the Flask Python framework, Apache server and the surrounding ecosystem of third-party libraries for the implementation.

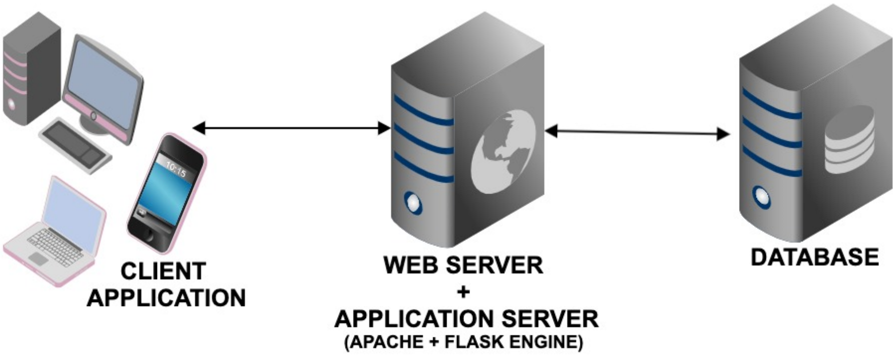


Fig. 6.1 An illustration of VAPA client-server architecture

6.3.1 Front-end or Client-side

The Front-end is considered the client-facing side of the system. Its primary role is to interface with the server and convey the information provided by the user. The front-end interface accepts requests, conducts various operations, and dynamically displays the results in tables and HTML formatted reports. The client front-end provides the main toolbar that allows end-users to interact with the system and the required data analysis. The core tools that are used in the process are HyperText Markup language (HTML), Cascading Style Sheets (CSS) and JavaScript (JS).

6.3.1.1 HTML

HTML belongs to a group of markup languages [69], which defines the overall structure of the page. These languages give web browsers instructions on rendering and managing web content. Instructions are expressed using elements or tags to display graphics, reference hyperlinks, and format textual data. Browsers translate HTML into formatted text displayed on a computer screen or mobile device.

6.3.1.2 CSS

CSS is responsible for the styling of the existing page structure. It was fundamentally designed to enforce the separation of webpage content from webpage style, including features such as fonts, colours and layout. This separation enables several HTML documents to share the style specified in a separate ".css" file. CSS can be written independently, thus making it possible to store it outside of HTML files. It, therefore, offers separation of concerns as style and structure are not coupled together.

6.3.1.3 JS

JavaScript is an internet client-side script that is supported by all major web browsers, including Internet Explorer, Chrome, and Firefox [87]. It's an interpreted and lightweight, object-oriented programming language. It allows the development of a dynamic website with an enhanced user interface. JavaScript is open source and can be embedded within an HTML page and provide cross-browser support. In addition, JavaScript is useful when validating data in HTML forms before sending them to a server. [87]. Furthermore, Asynchronous JavaScript and XML (AJAX) is one of the most valuable features of JS; this technique may be used to send and receive from a server all the required data without refreshing the webpage. Using DataTables, a powerful JS library, we add interactive features to all the HTML tables.

6.3.2 Backend

The backend is the server-side that focuses primarily on web application logic or how the programme operates. It is the process of building the web application's core, developing its platform, and populating it with all of the essential functionality. The server-side handles the data from the front-end and returns the result in a format that the client-side can understand. The server side consists of three main parts: web server software, application logic and a database.

6.3.2.1 Web Server Software

A *web server software* is a programme that runs on hardware and serves data to clients, which are typically browsers. Web server software comprises various components, the most important of which is an HTTP server. It is software that understands the HTTP protocol and web addresses [69]. The content of the hosted websites is sent to the end user's device through an HTTP server, which can be accessed by the domain names of the websites it stores. Whenever a browser needs a file hosted on a web server, the browser requests the file via HTTP. The HTTP server then accepts the request, finds the requested document, and sends it back to the browser.

The web server is implemented with the Apache HTTP Server.[1] Apache HTTP Server is a freely available open-source code implementation of an HTTP server. This server is part of the Apache Software Foundation and was developed by volunteers. While there are other alternative HTTP servers, Apache is used due to its simplicity. Additionally, Apache is well documented and has a lot of support and tutorials available on the internet. In addition, the mod WSGI module was used together with the Apache server as we need a WSGI compliant interface to host the flask (Python) application under Apache.

6.3.2.2 Application Logic

Application logic is also called the server's business logic and controls an application's functionality. It is responsible for all operations related to processing the requested and sent data, saving the data to the database, deciding what is required, and querying the database for the required data. Application logic is one of the most crucial parts of backend development that prescribes how the data can be displayed, created, saved and manipulated. It supports the implementation of core functionality such as session management, authentication mechanisms, output formatting, database interaction. Flask framework is used in this project due to its simplicity and flexibility. It is built on Python and supports several packages. This part is where the ML models of SkillVet are leveraged to make traceability predictions.

6.3.2.3 Database

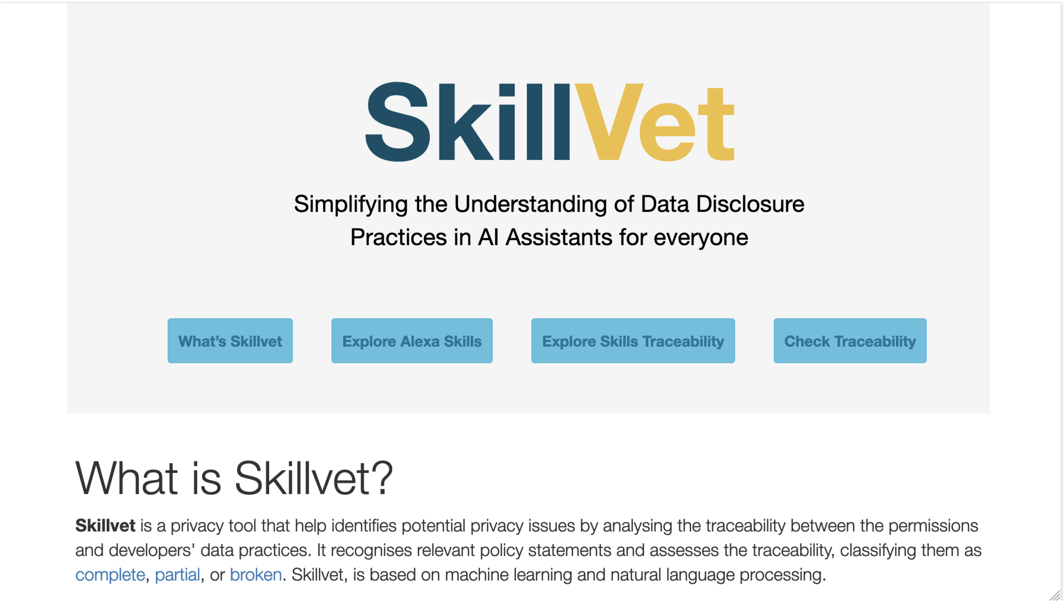
This part is responsible for storing the information in this architecture. In addition, this layer maintains and shares databases for the application. It enables the application logic layer to query (retrieve) data from the database and process it further. In the design phase, all the requirements were analysed and based on the needs, a database was designed to store just some of the information for the web application. Therefore, no user data will be collected, and thus, no regular backup of the database is necessary. The database only consists of one table that has two fields. The two fields in the table are designed to store strings. The values that can be expected in the type field are the permissions and privacy policy text, and they each have a corresponding value in the data field. This is processed at run-time to provide the relevant information to the application logic.

[1] https://httpd.apache.org/

6.4 System Implementation

VAPA is implemented as a graphical user interface (GUI) application. Using GUI offer a simple and understandable design architecture, making interaction with the application natural and approachable. However, designing VAPA as a GUI application also implies that the users never see what is beneath the GUI. Therefore what they see reflects everything the application can do. As a result, the user interface must inform them of everything they need to know about the application to utilise its capabilities thoroughly. Furthermore, the application uses a Single Page Application (SPA), eliminating the need to load all resources for each user's interaction. Overall, the application consists of five essential parts.

1. The default page/section which introduces VAPA features and functionality to the users
2. The documentation section that explains how VAPA works
3. The statistics section to explore the general data for the Amazon Alexa ecosystem
4. The traceability section that lets users explore and check skill data practices
5. Lastly, the section to explore the traceability results of skills longitudinally across years



Simplifying the Understanding of Data Disclosure Practices in AI Assistants for everyone

What is Skillvet?

Skillvet is a privacy tool that help identifies potential privacy issues by analysing the traceability between the permissions and developers' data practices. It recognises relevant policy statements and assesses the traceability, classifying them as complete, partial, or broken. Skillvet, is based on machine learning and natural language processing.

Fig. 6.2 VAPA homepage in laptop mode

6.4.1 Home Page

This is the first section encountered by the user when they visit the application. It introduces the user to the application's features and functionality. It consists of four tabs to help users navigate the site. These tabs link to the other essential sections on the web application. Figure 6.2 shows the default page/section when a user visits the web application link at https://skillvet.nms.kcl.ac.uk/ using their laptop.

6.4.2 VAPA Documentation

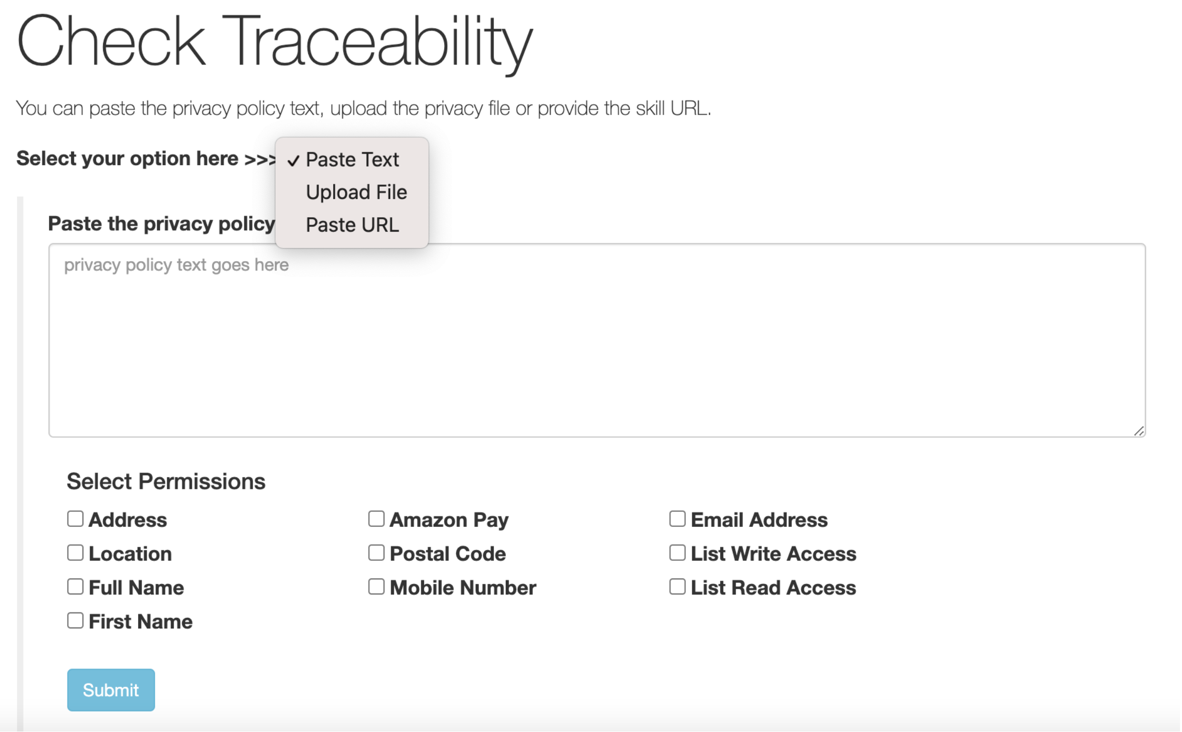
The web application interface needs to have the breadth of information necessary for it to be useful. This information needs to be relevant, not to distract users from the information they need. While we ensure that the application interface is simple enough not to warrant any further explanation, we also provide documentation to help users understand how to complete their tasks. Users must be aware of how their activities will be reflected in the app. The documentation section is where we disclose how the application works. We explain how the traceability analysis is done so users can understand the application results. Additionally, it shows links to related publications and this project's source code.

6.4.3 Alexa Statistics

To provide users with general information about the skill ecosystem, we include this section to lets users explore the number of skills and developers. This section will also let users understand how skills and developers have evolved. Using tables, this section offers information regarding the number of skills and developers found across the years.

6.4.4 Checking Skill Traceability

We implement a section that lets users check skill traceability. As shown in Figure 6.3, the traceability section accepts user input pertaining to skill attributes and saves it to the database. This lets users interested in checking skill traceability submit the skill privacy policy and permission. All of the inputs are implemented with HTML form. Validations have been used in the forms to check if the user has filled in all the required fields. If the user submits the form without filling out all or any of the required fields, they will be prompted to do so by displaying the instruction “Please fill in this field”. After filling the required field, the user can select the submit button. This sends the data to the backend, where it is saved in the database and used to process the skill traceability. After the application successfully processes the request, the user is redirected to the traceability result section (see Figure 6.13).



Check Traceability

You can paste the privacy policy text, upload the privacy file or provide the skill URL.

Select your option here >>> ✓ Paste Text / Upload File / Paste URL

Paste the privacy policy

privacy policy text goes here

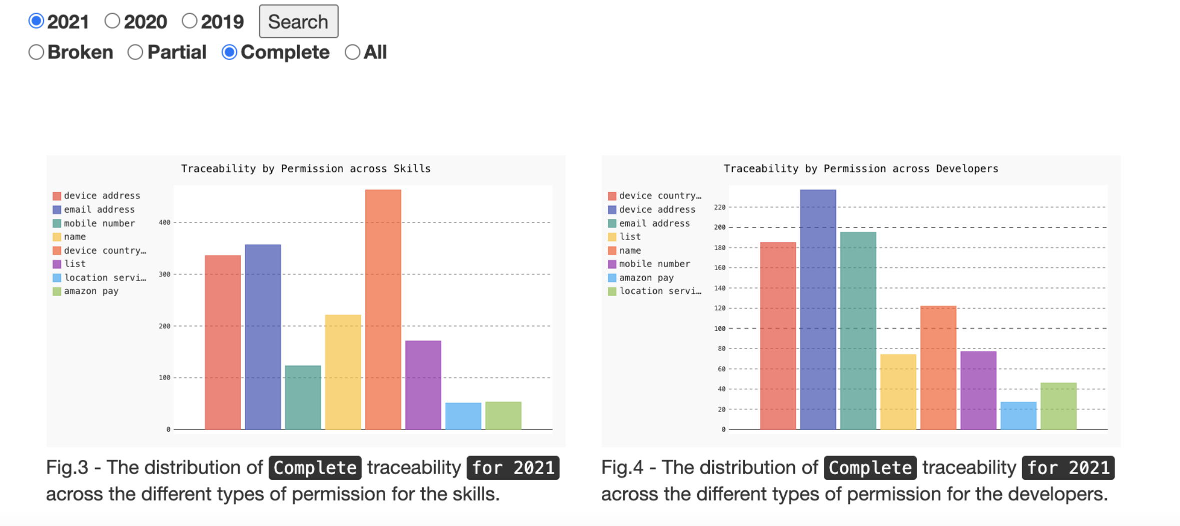
Select Permissions

□ **Address**
□ **Location**
□ **Full Name**
□ **First Name**
□ **Amazon Pay**
□ **Postal Code**
□ **Mobile Number**
□ **Email Address**
□ **List Write Access**
□ **List Read Access**

Fig. 6.3 VAPA traceability section showing the different traceability options

6.4.5 Explore Skill Traceability Longitudinally

To allow users to explore traceability results of skills longitudinally across the years, we implemented a section that let users explore how traceability has changed over time and understand what brought about these changes. Using charts and tables, this section offers information regarding skills traceability by type of permissions, skill traceability by categories, skill traceability by skills requesting permission information, and skill traceability by the number of skills & developers. These will offer users vital insight into how the ecosystem has evolved regarding third-party developers' privacy practices. One good example of the chart is shown in Figure 6.4 where users can view the skill traceability by type of permissions over time.



● 2021 ○ 2020 ○ 2019 Search

○ Broken ○ Partial ● Complete ○ All

Fig.3 - The distribution of `Complete` traceability `for 2021` across the different types of permission for the skills.

Fig.4 - The distribution of `Complete` traceability `for 2021` across the different types of permission for the developers.

Fig. 6.4 VAPA traceability by type of permissions

6.5 Evaluation

Here, we evaluate the developed VAPA application to the requirements given in Section 6.2. The requirements are divided into two groups. The first one is about the technical aspects of the application. The second set of requirements are features that need to be implemented in the application as part of the project.

6.5.1 Technical Requirements

6.5.1.1 Coherent Visual Design

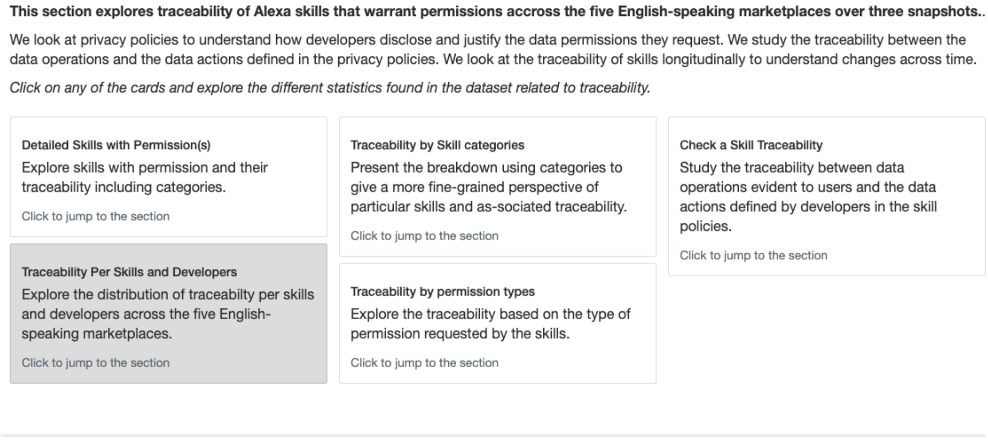
In order to speed up the development process, it is recommended to use a CSS framework. We use Bootstrap CSS framework[2] being the most popular. One good advantage of using a framework is that it ensures a coherent look throughout the application. CSS framework allows a developer to build good looking applications without in-depth knowledge of designing techniques. It also provides responsive design out of the box and saves time by providing ready-to-use building blocks.

6.5.1.2 Intuitive Navigation

This is implemented with a distinctive background colour and shape. It is vital to have an easily accessible navigation system since the user needs a fast way of transitioning between different application sections. On the other hand, if the navigation is complicated

[2] https://getbootstrap.com/

or confusing, users could find it challenging to use it. That is why clickable cards and tabs are placed in positions where it is immediately visible, thus serving as a navigation map for the entire application (See Figure 6.5).



This section explores traceability of Alexa skills that warrant permissions accross the five English-speaking marketplaces over three snapshots..

We look at privacy policies to understand how developers disclose and justify the data permissions they request. We study the traceability between the data operations and the data actions defined in the privacy policies. We look at the traceability of skills longitudinally to understand changes across time.

Click on any of the cards and explore the different statistics found in the dataset related to traceability.

Detailed Skills with Permission(s)
Explore skills with permission and their traceability including categories.
Click to jump to the section

Traceability Per Skills and Developers
Explore the distribution of traceabilty per skills and developers across the five English-speaking marketplaces.
Click to jump to the section

Traceability by Skill categories
Present the breakdown using categories to give a more fine-grained perspective of particular skills and as-sociated traceability.
Click to jump to the section

Traceability by permission types
Explore the traceability based on the type of permission requested by the skills.
Click to jump to the section

Check a Skill Traceability
Study the traceability between data operations evident to users and the data actions defined by developers in the skill policies.
Click to jump to the section

Fig. 6.5 Clickable cards for easy navigation

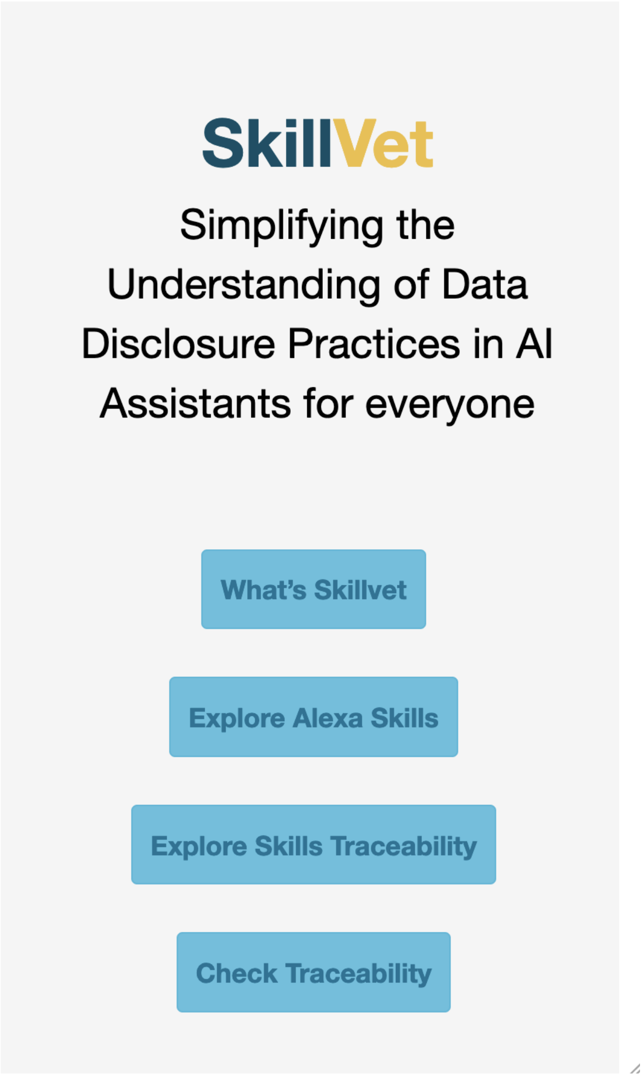
6.5.1.3 Responsive Design

The website is compatible with modern browsers and supports multiple screen sizes with a clear call to action on its purpose and uses. This behaviour is an essential part of any modern web application since access from mobile and tablet devices accounts for half of all the internet traffic in the world. Responsive design in the VAPA application allows the user to interact with it comfortably. The VAPA application in Laptop mode and mobile mode can be observed in Figure 6.2 and Figure 6.6 respectively. We can see that irrespective of the screen size/devices, the users will have no restrictions in accessing the information contained on the web page.

6.5.1.4 Fast Performance

Fast performance is a fundamental characteristic of modern web applications. It directly affects the overall user experience. For example, a web application with long load times could result in a significant user drop-off. Besides the initial loading time, the performance during the work of the application is also essential. Drop-in frame rate, stutter scrolling, and slow content loading are all issues that could influence the users' experience negatively. Besides, the application needs to present feedback to the user as quickly as possible.

The performance of the VAPA application has been ensured via modern web development techniques and tools. By following the best web development practices, we

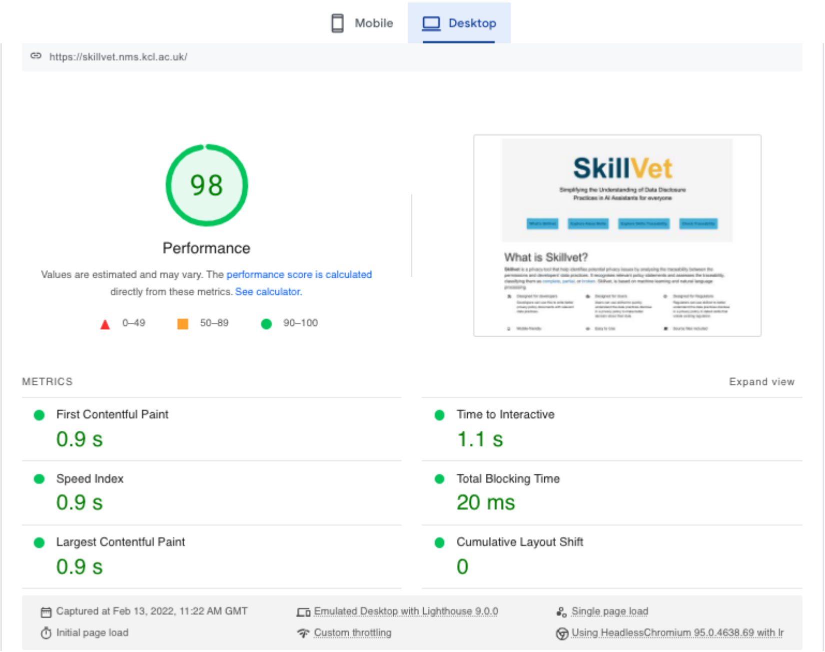


SkillVet

Simplifying the Understanding of Data Disclosure Practices in AI Assistants for everyone

Fig. 6.6 VAPA homepage in mobile mode

accomplished this particular requirement with good results. The performance metric is tested via a unique development tool maintained by Google called Lighthouse [86]. It runs several audits against the page and forms the report with summarised results. The developed VAPA application has been tested with Lighthouse, and the result is shown in Figure 6.7 for desktop devices and Figure 6.8 for mobile devices. The overall score is 98 out of 100 for desktop devices, where all the six performance metrics show excellent time results. Equally, we obtain 65 out of 100 for mobile devices. However, only one metric out of six shows excellent time results. Notwithstanding, the audit result reveals that the core web vitals assessment is passed, indicating delivery of fast and (what Google calls) delightful experience to visitors.



Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

METRICS

● First Contentful Paint
0.9 s

● Time to Interactive
1.1 s

● Speed Index
0.9 s

● Total Blocking Time
20 ms

● Largest Contentful Paint
0.9 s

● Cumulative Layout Shift
0

Captured at Feb 13, 2022, 11:22 AM GMT
Emulated Desktop with Lighthouse 9.0.0
Single page load
Initial page load
Custom throttling
Using HeadlessChromium 95.0.4638.69 with lr

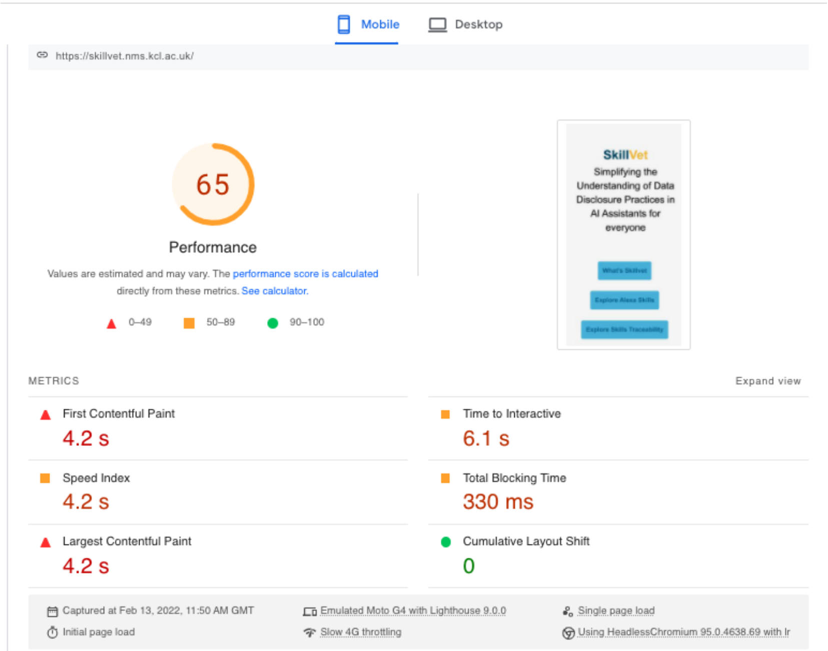
Fig. 6.7 Lighthouse performance audit results for desktop devices

To understand the reason for the poor performance in mobile devices, we use webpagetest[3] to have a detailed waterfall view of the web application connection. As shown in Figure 6.9, a total time of 8.5 secs is used to load the web application completely. However, we could see that 44% (3.75 sec) of this time was spent loading the JavaScript libraries and associated modules. Hence, the poor performance stems from the extended initial loading time. We could attribute this to the use of a single-page app design as the page is requested from the server in a single request, and therefore, there is a need to load all the resources at once.

6.5.1.5 Code Re-usability and Maintainability

Code reusability and maintainability are crucial aspects to consider in software development due to their knock effects. It saves development time for future projects and reduces technical errors. Although, not all code is reusable. Nevertheless, it is essential to ensure that different code components should be made as simple as possible to increase the likelihood of them being reused. Therefore, we implement our code following the SOLID design principles [146]. For instance, our application is implemented based on the single responsibility principle where each module performs one responsibility. This helps reduce the code dependencies, complexity and prevents any unexpected side-effects

[3] www.webpagetest.org



Mobile Desktop

https://skillvet.nms.kcl.ac.uk/

65

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

SkillVet
Simplifying the Understanding of Data Disclosure Practices in AI Assistants for everyone
What's Skillvet
Explore Alexa Skills
Explore Skills Traceability

METRICS Expand view

▲ First Contentful Paint
4.2 s

■ Time to Interactive
6.1 s

■ Speed Index
4.2 s

■ Total Blocking Time
330 ms

▲ Largest Contentful Paint
4.2 s

● Cumulative Layout Shift
0

Captured at Feb 13, 2022, 11:50 AM GMT
Initial page load
Emulated Moto G4 with Lighthouse 9.0.0
Slow 4G throttling
Single page load
Using HeadlessChromium 95.0.4638.69 with lr

Fig. 6.8 Lighthouse performance audit results for mobile devices

of future changes. Likewise, to fulfil the open-close principle, our modules are written so that any modification to enhance its functionality will only require adding new code rather than altering existing code. This makes the code to be reusable, extendable, but not modifiable.

Furthermore, the modularisation of components also follows the DRY principle in software engineering as it aims to reduce the repetition of code within the codebase. This once again helps the maintainability of the code as it reduces the size of the codebase, allowing it to be more readable and therefore making it easier to implement changes that need to occur during the production lifecycle of the code.

6.5.1.6 Handling Exceptions

The user must be aware of the application's state while running and need to be informed about what is going on through appropriate feedback. This is especially important when something goes wrong. For example, when the user makes a selection and the outcome they anticipated does not occur, they need to be aware of what is wrong and potentially why. This type of open and continuous communication could help build user trust in the web application. However, the user must only be notified if there has been an error. Such error messages need to be expressed in ways that precisely indicate the problem and constructively suggest a solution. It is not worth informing the user of the success

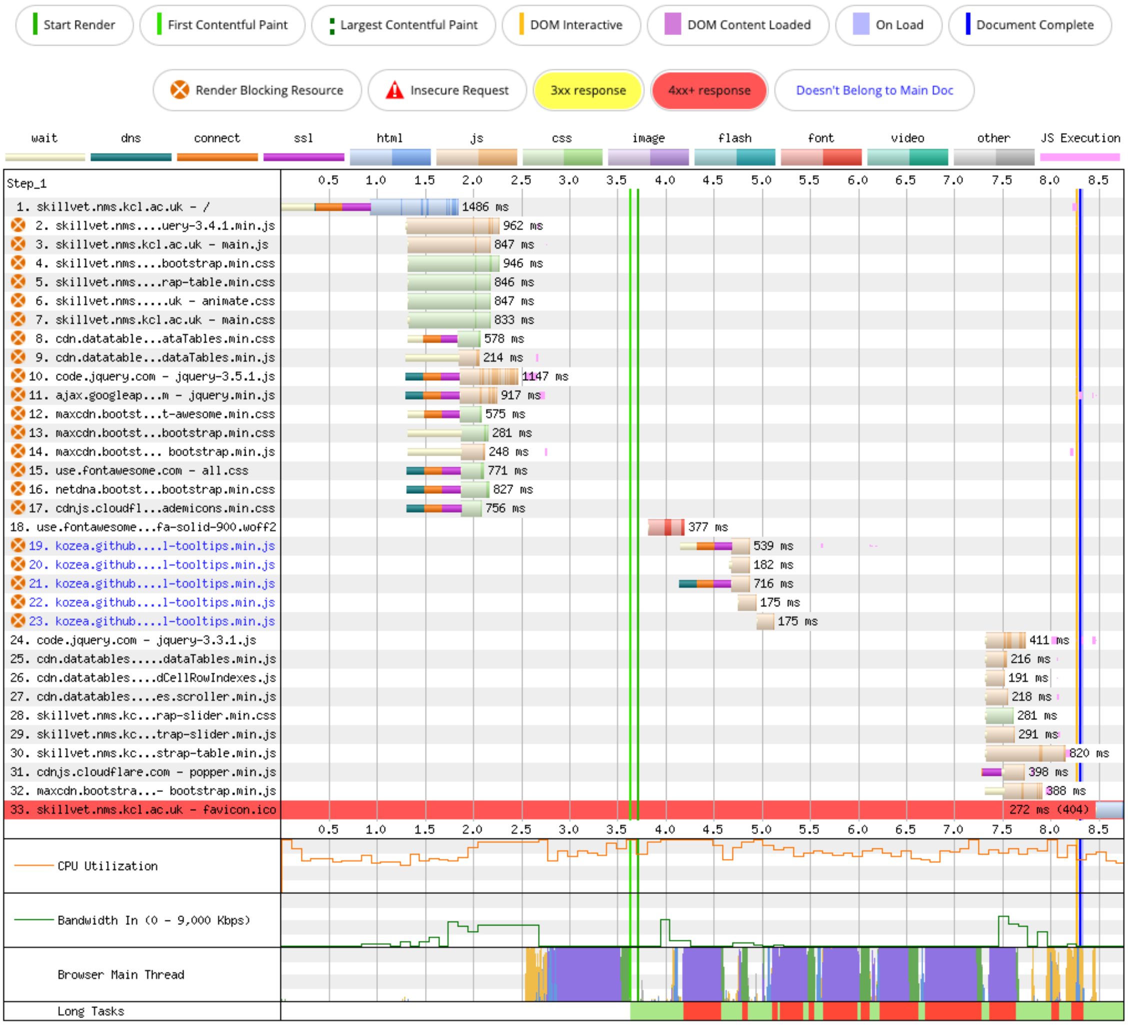

Start Render
First Contentful Paint
Largest Contentful Paint
DOM Interactive
DOM Content Loaded
On Load
Document Complete
Render Blocking Resource
Insecure Request
3xx response
4xx+ response
Doesn't Belong to Main Doc
wait
dns
connect
ssl
html
js
css
image
flash
font
video
other
JS Execution
Step_1
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
1. skillvet.nms.kcl.ac.uk - / 1486 ms
2. skillvet.nms....uery-3.4.1.min.js 962 ms
3. skillvet.nms.kcl.ac.uk - main.js 847 ms
4. skillvet.nms....bootstrap.min.css 946 ms
5. skillvet.nms....rap-table.min.css 846 ms
6. skillvet.nms.....uk - animate.css 847 ms
7. skillvet.nms.kcl.ac.uk - main.css 833 ms
8. cdn.datatable...ataTables.min.css 578 ms
9. cdn.datatable...dataTables.min.js 214 ms
10. code.jquery.com - jquery-3.5.1.js 1147 ms
11. ajax.googleap...m - jquery.min.js 917 ms
12. maxcdn.bootst...t-awesome.min.css 575 ms
13. maxcdn.bootst...bootstrap.min.css 281 ms
14. maxcdn.bootst... bootstrap.min.js 248 ms
15. use.fontawesome.com - all.css 771 ms
16. netdna.bootst...bootstrap.min.css 827 ms
17. cdnjs.cloudfl...ademicons.min.css 756 ms
18. use.fontawesome...fa-solid-900.woff2 377 ms
19. kozea.github....l-tooltips.min.js 539 ms
20. kozea.github....l-tooltips.min.js 182 ms
21. kozea.github....l-tooltips.min.js 716 ms
22. kozea.github....l-tooltips.min.js 175 ms
23. kozea.github....l-tooltips.min.js 175 ms
24. code.jquery.com - jquery-3.3.1.js 411 ms
25. cdn.datatables.....dataTables.min.js 216 ms
26. cdn.datatables....dCellRowIndexes.js 191 ms
27. cdn.datatables....es.scroller.min.js 218 ms
28. skillvet.nms.kc...rap-slider.min.css 281 ms
29. skillvet.nms.kc...trap-slider.min.js 291 ms
30. skillvet.nms.kc...strap-table.min.js 820 ms
31. cdnjs.cloudflare.com - popper.min.js 398 ms
32. maxcdn.bootstra...- bootstrap.min.js 388 ms
33. skillvet.nms.kcl.ac.uk - favicon.ico 272 ms (404)
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
CPU Utilization
Bandwidth In (0 - 9,000 Kbps)
Browser Main Thread
Long Tasks

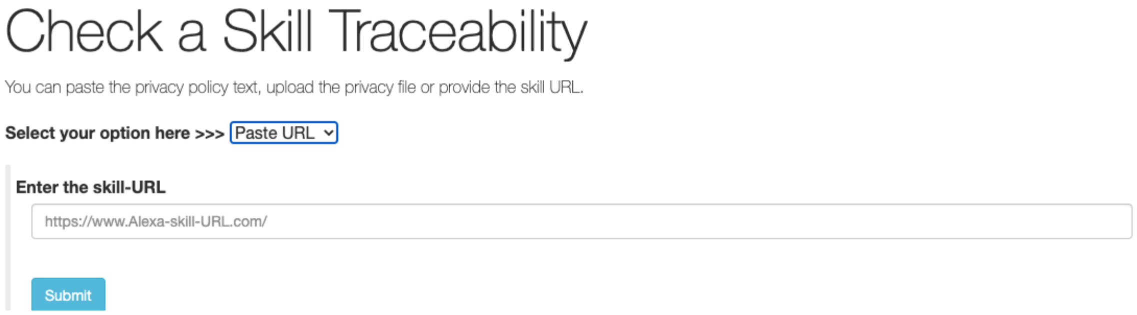
Fig. 6.9 Webpagetest connection audit results

if their desired outcome is visible on screen. In addition, it is helpful to display feedback to the user of success if they have made a change that is not visible on screen. One good example of the features implemented to meet this requirement is the form validation earlier discussed in Section 6.4.4 that notifies users when the form in the traceability section is not filled. As VAPA supports uploading different file formats, users are also notified when an unsupported file format is selected.

6.5.2 VAPA Features Requirement

6.5.2.1 Traceability Checker

To check the skill traceability, users can upload the skill privacy policy file or paste the privacy policy text in the text area box provided on the traceability page and select relevant permissions requested by the skills. VAPA supports uploading multiple file extensions such as pdf, txt, Docx, HTML, and image extensions like jpeg, jpg, gif, and png. This will let users specify the data collected and offer different choices when uploading privacy policy documents.



Check a Skill Traceability

You can paste the privacy policy text, upload the privacy file or provide the skill URL.

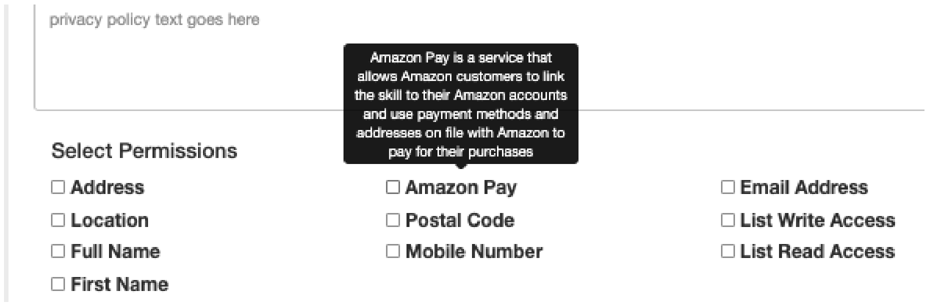
Select your option here >>> Paste URL

Enter the skill-URL

https://www.Alexa-skill-URL.com/

Fig. 6.10 Checking traceability via skill URL

VAPA also support traceability analysis via skill URL (see Figure 6.10). To do this, we implement a web crawler that systematically visits the skill website to fetch the skill requested permissions and privacy policy needed for the traceability analysis. In addition, by making the dropdown choices simple, users can easily switch between checking skill traceability options. As shown in Figure 6.3, users have the option to check whether to paste text data, upload a file or paste a URL link. Furthermore, we added a hover text to each of the permission elements (see Figure 6.11). This text contains information describing the meaning of the selected permission when the user interacts with them to help users with their choices.



privacy policy text goes here

Amazon Pay is a service that allows Amazon customers to link the skill to their Amazon accounts and use payment methods and addresses on file with Amazon to pay for their purchases

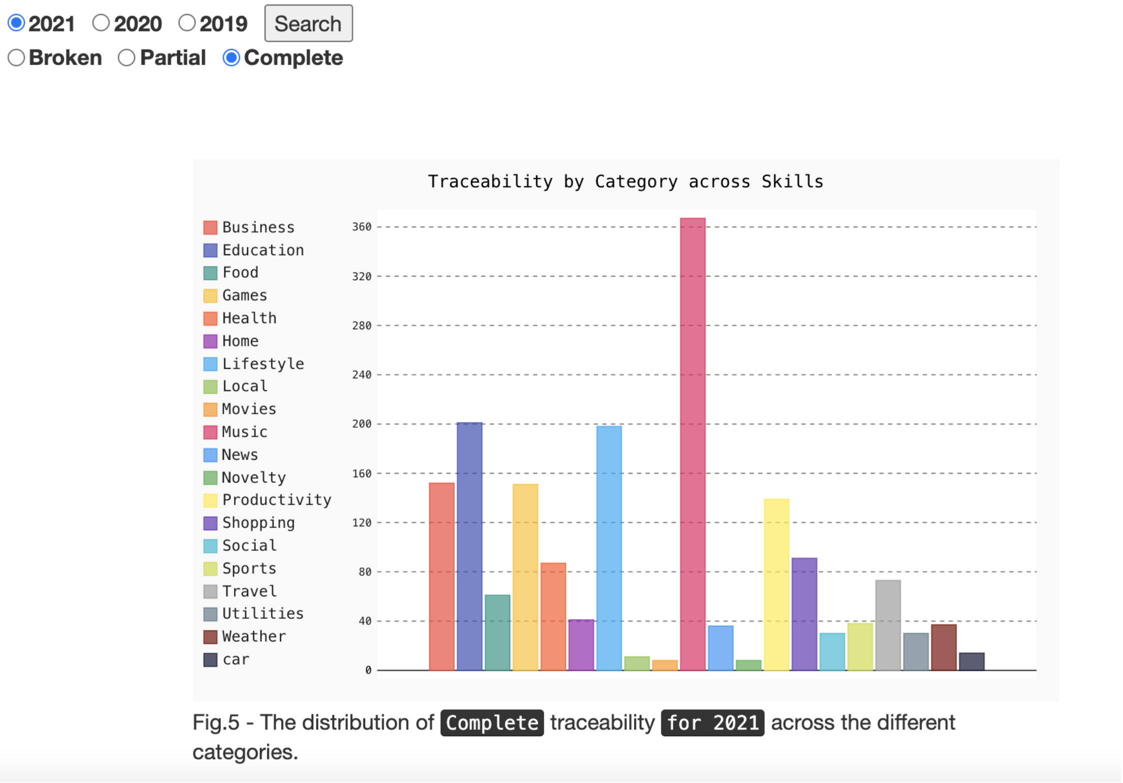
Select Permissions

□ Address
□ Location
□ Full Name
□ First Name
□ Amazon Pay
□ Postal Code
□ Mobile Number
□ Email Address
□ List Write Access
□ List Read Access

Fig. 6.11 Hovered *Amazon Pay* permission showing information describing its meaning

6.5.2.2 Interactive Statistical Display

We include interactive charts and tables to address an interactive statistical display requirement. This we accomplished using DataTables[4] which is a powerful JS library and Pygal library.[5] The DataTables and Pygal libraries let us add interactive features to all the HTML tables and charts, respectively. Furthermore, the interactive display also supports filtering statistical information by year of interest. For example, Figure 6.12 shows the VAPA Traceability by category interactive screen. Users can filter the result by selecting the year they want to view and the type of traceability they are interested in. Likewise, they can interact with the chart by clicking on the bar to see the corresponding figure. Users can also click on the legend to hide or un-hide a category.



● 2021 ○ 2020 ○ 2019

○ Broken ○ Partial ● Complete

Fig.5 - The distribution of Complete traceability for 2021 across the different categories.

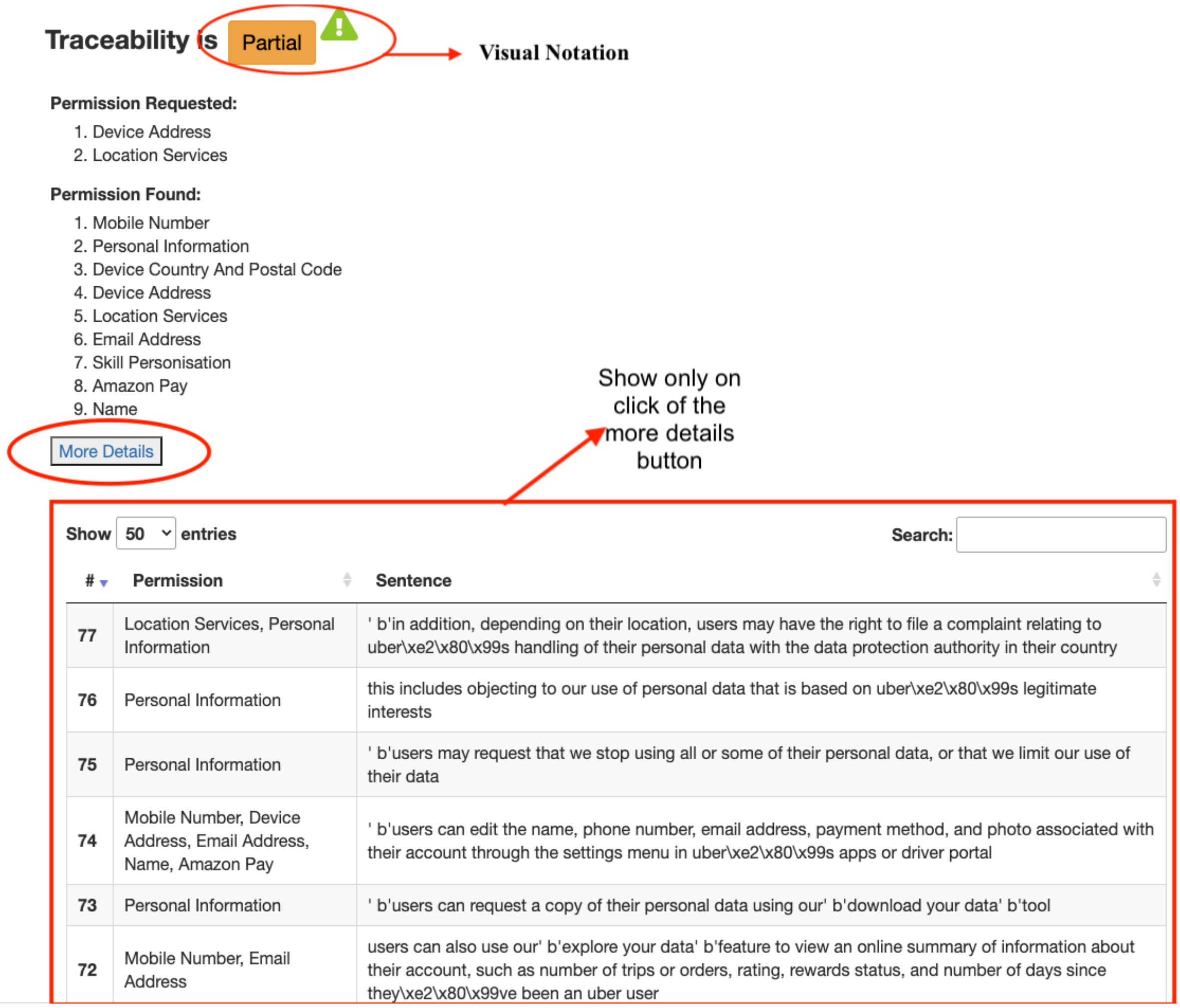
Fig. 6.12 VAPA traceability by category interactive section

6.5.2.3 Traceability Result Display

We implement a result display section to provide users with information about the result of the traceability check. This section of the web page shows the permissions requested

[4] https://datatables.net/

[5] http://www.pygal.org/en/stable/



Traceability is Partial

Visual Notation

Permission Requested:

1. Device Address
2. Location Services

Permission Found:

1. Mobile Number
2. Personal Information
3. Device Country And Postal Code
4. Device Address
5. Location Services
6. Email Address
7. Skill Personisation
8. Amazon Pay
9. Name

Show only on click of the more details button

#	Permission	Sentence
77	Location Services, Personal Information	' b'in addition, depending on their location, users may have the right to file a complaint relating to uber\xe2\x80\x99s handling of their personal data with the data protection authority in their country
76	Personal Information	this includes objecting to our use of personal data that is based on uber\xe2\x80\x99s legitimate interests
75	Personal Information	' b'users may request that we stop using all or some of their personal data, or that we limit our use of their data
74	Mobile Number, Device Address, Email Address, Name, Amazon Pay	' b'users can edit the name, phone number, email address, payment method, and photo associated with their account through the settings menu in uber\xe2\x80\x99s apps or driver portal
73	Personal Information	' b'users can request a copy of their personal data using our' b'download your data' b'tool
72	Mobile Number, Email Address	users can also use our' b'explore your data' b'feature to view an online summary of information about their account, such as number of trips or orders, rating, rewards status, and number of days since they\xe2\x80\x99ve been an uber user

Fig. 6.13 VAPA detailed result page

by the skills and the identified data practices found in the skill privacy policy document. As we see in Figure 6.13, the test skill requested for *Device Address* and *Location Services* permissions. However, only *Device Address* could be found in the privacy policy document, hence, the reason for the partial traceability result. Not only do we display our result in text, but we also attempt to show it using a visual notation. Depending on the traceability result, the visual notification could be red (broken traceability), yellow (partial traceability) and green (complete traceability).

We extend this further by adding a button that allows users to see an interactive table with a detailed breakdown of the traceability result. As shown in Figure 6.13, VAPA displays on click of the “More Details” button a table with the statements that correspond to the data practices identified in the skill privacy policy. This helps elucidate how VAPA maps permission(s) to policy statements.

6.5.2.4 Longitudinal Traceability Result

We implement this requirement to allow users to explore traceability results of skills longitudinally across years to track the evolution of the ecosystem. This requirement is fulfilled by leveraging the measurement data from our characterisation of all Alexa skills across the five English-speaking countries over three years (c.f Section 5). As a result, users can explore how traceability has changed from 2019 to 2021. Using charts and tables, users can explore longitudinal traceability information by type of permissions, categories, number of skills, and number of developers. For instance, Figure 6.12 shows the VAPA Traceability by category, where users can explore how traceability has changed across the different skill categories.

6.6 Discussion

We have developed the application front-end using HTML, CSS and JS, and the Server-side leveraged Flask framework and Apache server. We separated the web application code into different modules, making it clean, reusable, easy to maintain and extend in the future.

To provide a positive user experience, our implementation follows established platform and industry conventions without the need for users to think whether different words, or actions mean the same thing. The interface is designed to keep users informed about what is going on through appropriate feedback. Also, we try to simplify the expressions used on the web page so that users can understand them without any difficulty. Although we ensure that the application interface is simple enough not to warrant any further explanation, we provide documentation to help users understand how to complete their tasks and ensure that it does not contain irrelevant information which could distract them from the information they need.

Users often perform actions by mistake. We make sure we eliminate error-prone conditions and present users with a confirmation option before committing to an action. When necessary, we express error messages in ways that precisely indicate the problem and constructively suggest a solution. In addition, users also need a way to address these mistakes without going through an extended process. When users can back out of a process or undo an action, it fosters a sense of freedom and confidence. Our app interface implementation is simple, making it easy for users to undo and redo their actions, giving them a sense of freedom.

For flexibility and efficiency of use, we implemented features that cater to inexperienced and experienced users. For example, the traceability checker supports customisation as users can choose how they want to work. Likewise, the result display feature lets users decide what level of details they need. We also design the app interface to promote recognition and reduce the information that users have to remember. Likewise, we make sure that information required to use the app interface are visible when needed.

Testing can be used to benchmark how an application performs against the expectations of the developer and the client. It is conducted to see if the application was functional from a user perspective. In this project, we performed developmental testing, comprising system testing, unit testing and compatibility testing. However, due to time constraints, we did not perform user testing. Testing the application with users could help uncover users' familiar terminology, as well as their mental models around important concepts. Of course, the application functionality was continuously tested manually. Nevertheless, since the development team was too close to the project, independently assessing the application's functionality was not easy as we knew how it worked. Therefore, we hope to recruit participants and test the application with users further to evaluate the application's usability as future work. This will let us add new features and revise them according to user requirements.

Equally, VAPA is implemented as a "single-page application". Single-page application is a prevalent approach nowadays [80], and it has many advantages. For instance, single-page applications: 1) download the page whenever it is executed without hitting the server each time the client interacts with the application 2) it offers smooth navigation and fast speeds after loading 3) it provides users with a simple linear experience 4) are compatible with offline services when an internet connection is lost. However, when evaluating the application performance, a major problem with single-page apps is their long initial loading time. The page is requested from the server in a single request and, therefore, needs to load all the resources at once.

While our implementation broadly covers the ten Nielsen's usability heuristics for user interface design [158], there is an opportunity for improvement. Based on the heuristics, we could further provide better matching between the application and the real world using words, phrases, and concepts familiar to the users. Furthermore, as with any software project, other technologies and approaches could still be used to implement this application. Notwithstanding, as seen from the requirements evaluation in Section 6.5, we believe the methods and tools used are sufficient to meet the application-specific requirements and fulfil the project objective.

6.7 Conclusion

Evaluating application traceability could assist in determining whether the transparency and privacy control mechanisms provided to users are adequate to protect their privacy. In this chapter, we presented VAPA, a single page GUI privacy tool based on SkillVet that could be used to evaluate skill traceability and identify potential privacy issues in these voice-driven applications. VAPA could be valuable when detecting inappropriate usage of data permissions. This could be leveraged by Skill developers to write better privacy policy documents with relevant data practices. More importantly, irrespective of a user's technical background, VAPA could be used to understand the data practices disclosed in a privacy policy, allowing them to be more aware of the skill data practices. In addition, regulators could use it to detect skills that violate existing privacy regulations.

Chapter 7

Conclusions and Future Work

We provide concluding remarks for the work presented in this thesis in Section 7.1, and points to some interesting directions for future work in Section 7.2.

7.1 Conclusions

SPA have become very popular systems mostly due to their interactive technology. This allows users to interface with networked appliances easily and consume all kinds of online services using natural language. The work in this thesis has looked at the security and privacy issues of SPA.

We began by looking at the generic architecture of the SPA and identified the essential points that are important to understand potential weaknesses. We showed that several elements expose these assistants to various risks, such as the complexity of their architecture, the AI features they rely on, and their use of a wide range of underlying technologies. We highlighted the assets in the architecture from SPA users' point of view. We discussed why users consider the assets essential to understand what is at stake and what should be protected. Furthermore, we conducted a comprehensive analysis of existing security and privacy attacks on these assets and the current countermeasures to mitigate these attacks. By mapping the attacks and countermeasures to the SPA architecture, we found out that while the attack surface of SPA is distinctly broad, the research community has focused only on a small part of it. In particular, recent works have mainly focused on issues related to the direct interaction between a user and their SPA. While those problems are indeed fundamental and further research is needed for effective countermeasures, we also found that research is required to address other issues related to authorisation, speech recognition, profiling, and the technologies integrated with SPA (e.g. the cloud, third-party skills, and other smart devices).

We conducted a systematic measurement study of SPA skills, identified as one of the architectural elements offering a broad attack surface and needing more attention. We performed the first large-scale analysis of the Amazon Alexa skill ecosystem — the largest SPA ecosystem in terms of the number of skills by far, analysing over 199k third-party skills. We analysed the skills in the look for concerning privacy practice and study the extent of traceability between the data actions specified in the skill privacy policies and the related data operations obvious to users, evaluating the traceability as broken, partial or complete depending on how well the data practice is disclosed. Our findings uncovered bad privacy practices in about 43% of the Alexa skills that use permissions involving 50% of the developers with skills that request permissions. The results led to a responsible disclosure process where we reported 675 Alexa skills with privacy issues to Amazon and the affected developers.

There are a large amount of skills, and this number keeps increasing daily, making it challenging to perform thorough traceability analysis manually. To help with the traceability analysis at scale, we proposed SkillVet, an automated system that leverages NLP and ML to systematically understand a skill privacy policy and automate the accountability process. Given a skill, SkillVet first identifies and classifies all statements in its privacy policy that relate to data practices over personal information and maps each data statement with one or more permissions. SkillVet then compares these permissions with those the skill is authorised to request through the Amazon API during runtime. Depending if the permissions requested match those found in the policy, the skill is classified as having a complete, partial, or broken privacy policy. The system achieves 93% overall accuracy and, in particular, 99% accuracy for broken traceability. In addition, it can correctly differentiate and classify each of the permissions correctly, obtaining F1 scores and accuracy of over 90% for all data permissions.

Furthermore, to understand how the different attack vectors inherited from using third-party skills permeate through the markets, we performed a longitudinal measurement study of the Amazon Alexa skills. This measurement across three years allows us to answer novel research questions and show some key novel insights. In particular, we show that a) Amazon's skill vetting process improved over the years, though it is still not good enough; b) stricter scrutiny is given to skills that collect data via the Amazon API, but most skills that collect data via conversation (bypassing the API) do not disclose their data practices; c) skills can still be over-privileged even when they adequately disclose their data practices; d) most skills include a privacy policy only to fulfil Amazon requirements but do not promote awareness; e) several factors influence the traceability between stated and actual data practices, including changes in permissions; f) a responsible disclosure process like the one we did has a positive impact, with 356 out of 675 skills reported no longer posing a threat. All of these findings are crucial to informing third-party skill developers to improve their skills and for Amazon to continue improving further the vetting process.

Lastly, we implemented SkillVet as an online privacy assessment tool, providing users with an interactive visual editor that they can use to explore skill traceability regardless of their technical background. The tool allows the inspection of the traceability across years as well as to check the skill traceability live. It could help skills developers to better comply with privacy requirements regarding data practices as they could see whether they have adequately disclosed the requested personal data. Regulators and end-users can also leverage the tool to easily understand the data practices disclosed in a privacy policy and see if they violate any existing regulations. Notwithstanding, while the findings in this thesis could help contribute to the collective effort towards an ever more privacy-aware and secure SPA, there are still some exciting areas that this thesis does not cover. Therefore, in the next section, we discuss the future research direction that could further help consolidate our study.

7.2 Future Work

The research and contribution presented in this thesis raised different open questions. Therefore, in this section, we will be offering various ideas and methodologies, which can improve the results obtained in this work and inspire further investigations.

We suggest a number of future research directions. These include: i) improving SkillVet — the privacy assessment system propose in Chapter 4 of this Thesis, ii) implementing a better privacy policy delivery mechanism, iii) understanding developers' perceptions, mental models, and behaviours, or iv) expanding the current research scope. All of which are discuss in the following subsections

7.2.1 Improving SkillVet

We propose an automated system, SkillVet, that analyses skill traceability. However, SkillVet can only conduct the traceability analysis for English-speaking marketplaces, as this is the source language of our ground truth. While we could overcome this limitation using automated translation, this could introduce errors in the classification. A number of approaches have attempted to learn general-purpose multilingual representations (e.g., mBERT [244], XLM [129], XLM-R [53]), aiming to capture knowledge shared across languages to build systems that not only work with one language but across several languages. In terms of future work, it would be interesting to explore one of these approaches to train multilingual models for SkillVet and evaluate how these models perform with translated privacy policies or/and privacy policies written in other languages to support traceability analysis for non-English marketplaces.

Additionally, while SkillVet accounts for negations in a way proven effective for traceability analysis, future work should explore the use of ontologies as in [23]. Likewise, deep learning is currently not applicable to train our sentence classifiers as we do not have enough data, so a study exploring deep learning would be an interesting research direction. We further implement the dynamic interactive tool SkillExplorer reported in [88] to automatically interact with skills. However, SkillExplorer is restricted by design to skills with unique invocation utterances/names. We have seen in this paper that tens of thousands of skills share invocation names. Therefore, looking into methods to discover this type of skill, which are very prevalent, will add value to the present work. Importantly, however, SkillVet would easily integrate with any such tool when provided with the data the skills ask for and any accompanying privacy policies.

Furthermore, the account linking feature allows users to connect their identity with the one they use in a different system like Google, Amazon, or Facebook [16]. This enables the collection of further personal data without the need to use data permissions. Accounting for the exact data collected by developers through account linking is challenging because: 1) third-party sites used for account linking have all different formats and are thus challenging to scrape; 2) developers may ask for any type of personal information at any point, and not just during the registration. Therefore, a possible area of future research would be to develop an automated method to account for the actual data that skills collect using the account linking features. And then use this data together with the data collected by the skill from other means like API and conversation to look at the skill traceability.

7.2.2 Implementing a Better Privacy Policy Delivery Mechanism

The results obtained from the systematic review of skill permissions and corresponding privacy policies have shown that many privacy policies are not relevant, poorly written and not well-tailored to the skill data practices. Thus, they fail to help offer more transparency to the users regarding the skills data practices. Even when these policies are well written, the challenges of using different channels (web, compatible apps etc.) to deliver these policies instead of how the SPA service is offered (voice) may affect how consumers perceive the data practices executed as they use the service. Using different channels to deliver these policies could also make it challenging to access them, resulting in fewer users reading them. Failure to read privacy policies may lead to using privacy-sensitive applications without awareness of the data practices involved. Therefore, considerably more work will need to be done to study the content and delivery of privacy policies in SPA and see how they could embed within the user flow of interaction to ensure consistency [99, 200]. Further research efforts are needed to ensure that policy notices are relevant, actionable, well-tailored and understandable by the users.

7.2.3 Understanding Developers' Perceptions, Mental Models, and Behaviours

We have seen bad privacy practices in most developers (about 50%) with skills that request permissions. Particularly worrying cases are those in categories like Kids, Schools, and Education, which exhibit broken and partial traceability. Previous works [90, 162] in other domains have shown that developers struggle to program securely, and many have studied the challenges faced by the developers when it comes to secure software development[45]. Others have looked at the rationale underpinning developers' decisions, which eventually strengthen or weaken application security [234]. Therefore, understanding the challenges faced by the skill developers, understanding their mental models, perceptions, attitudes, and misconceptions about the skill development process will be interesting future research. Further work could study what these mental models are, identify which mental models are complete, incomplete, which ones leave the skill vulnerable, and whether such developers' attitudes and perspectives have important implications for SPA security and privacy.

7.2.4 Expanding the Current Research Scope

Our analysis focuses mostly on the Alexa skill ecosystem as it currently has, by far, the most substantial number of skills across categories [119] when compared with other well-known SPA like Google Assistant, Apple Siri and Microsoft Cortana. Nevertheless, while our findings apply to those other SPA as they all perform similar functions, share some common features, and have similar architecture, follow-up work on them would also be an exciting addition to the present work. Likewise, the present study has established a framework for uncovering inadequate data practices disclosure in the third-party voice application ecosystem. However, it will be interesting to see whether the methodology propose could be adapted to third-party text-based application ecosystems. Most notably, those similar to the skill ecosystem where third-party application also resides in the cloud and where source codes are not available to the public for analysis. For instance, the Slack, and Discord third-party ecosystems.

More importantly, our literature review has identified many open challenges in the SPA ecosystem. However, this thesis only focuses on one of them — the systematic measurement and privacy assessments of SPA skills, which is essential for assessing and mitigating the vulnerabilities inherent within the skill ecosystem offering a large attack surface. However, future research should look at other key identified issues such as strengthening authentication, enhancing authorisation models and mechanisms, building secure and privacy-aware speech recognition, developing AI-based security and privacy countermeasures, improving user awareness and usability, and studying further profiling attacks and defences. Furthermore, SPA security and privacy is a fast-moving field still in its infancy. For instance, the number of skills keeps growing daily, and could potentially usher in a new level of threats and threat actors. Besides, the SPA attack surface is vast, and the ever-growing integration of SPA with other IoT devices keeps widening the surface. Therefore, there is a need to continue expanding the understanding of the security and privacy issues in this domain to help prioritise the most promising areas and devise usable ways of improving them.

In conclusion, while the findings of this thesis have a number of important implications for future practice, there are still areas that are yet to be explored. To support other researchers interested in repeating and reproducing our work, we have made all our datasets publicly available, as well as our codes.

References

[1] Abdi, N., Ramokapane, K. M., and Such, J. (2019). More than smart speakers: Security and privacy perceptions of smart home personal assistants. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, Santa Clara, CA. USENIX Association.

[2] Abdi, N., Zhan, X., Ramokapane, K., and Such, J. (2021). Privacy norms for smart home personal assistants. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, pages 558:1–558:14.

[3] Aiyar, S. and Shetty, N. P. (2018). N-gram assisted youtube spam comment detection. *Procedia computer science*, 132:174–182.

[4] Alepis, E. and Patsakis, C. (2017). Monkey says, monkey does: Security and privacy on voice assistants. *IEEE Access*, 5:17841–17851.

[5] Alhadlaq, A., Tang, J., Almaymoni, M., and Korolova, A. (1902). Privacy in the amazon alexa skills ecosystem. *Star*, 217(11).

[6] Aloufi, R., Haddadi, H., and Boyle, D. (2019). Privacy preserving speech analysis using emotion filtering at the edge: Poster abstract. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, SenSys '19, page 426–427, New York, NY, USA. Association for Computing Machinery.

[7] Amar, Y., Haddadi, H., and Mortier, R. (2016). Privacy-aware infrastructure for managing personal data. In *Proceedings of the 2016 ACM SIGCOMM Conference*, SIGCOMM '16, page 571–572, New York, NY, USA. Association for Computing Machinery.

[8] Amazon (2017). About Alexa Voice Profiles. https://www.amazon.com/gp/help/customer/display.html?nodeId=202199440. [Online; last accessed 20-February-2019].

[9] Amazon (2018). The Alexa Skill Store for France is a Fast Growing Land of Opportunity. https://developer.amazon.com/docs/ask-overviews/understanding-the-different-types-of-skills.html. [Online; last accessed 29-December-2018].

[10] Amazon (2019a). Alexa Skills English Variants Migration. https://developer.amazon.com/es-ES/docs/alexa/faq/english-migration-faq.html. [Online; last accessed 24-September-2019].

[11] Amazon (2019b). All-new Echo Show (2nd Gen). https://www.amazon.com/All-new-Echo-Show-2nd-Gen/dp/B077SXWSRP. [Online; last accessed 7-January-2019].

[12] Amazon (2019c). Build for alexa and earn fantastic perks.

[13] Amazon (2019d). Conditions of use & sale. https://www.amazon.co.uk/gp/help/customer/display.html?nodeId=GLSBYFE9MGKKQXXM. [Online; last accessed June-2021].

[14] Amazon (2019e). 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. [Online; last accessed 16-December-2019].

[15] Amazon (2019f). Security Testing for an Alexa Skill. https://developer.amazon.com/docs/custom-skills/security-testing-for-an-alexa-skill.html. [Online; last accessed 03-July-2019].

[16] Amazon (2019g). Understand Account Linking. https://developer.amazon.com/docs/account-linking/understand-account-linking.html. [Online; last accessed 04-September-2019].

[17] Amazon (2019h). Understand In-Skill Purchasing. https://developer.amazon.com/docs/in-skill-purchase/isp-overview.html. [Online; last accessed 24-September-2019].

[18] Amazon (2020a). Build Skills with the Alexa Skills Kit. https://developer.amazon.com/en-US/docs/alexa/ask-overviews/build-skills-with-the-alexa-skills-kit.html. [Online; last accessed 15-October-2020].

[19] Amazon (2020b). Host a custom skill as a web service. https://developer.amazon.com/en-US/docs/alexa/custom-skills/host-a-custom-skill-as-a-web-service.html. [Online; last accessed 29-May-2020].

[20] Amazon (n.da). Develop Skills in Multiple Languages. https://developer.amazon.com/en-US/docs/alexa/custom-skills/develop-skills-in-multiple-languages.html. [Online; last accessed May-2020].

[21] Amazon (n.db). Understand How Users Interact with Skills. https://developer.amazon.com/en-GB/docs/alexa/ask-overviews/understanding-how-users-interact-with-skills.html. [Online; last accessed 21-February-2020].

[22] Ammari, T., Kaye, J., Tsai, J. Y., and Bentley, F. (2019). Music, search, and iot: How people (really) use voice assistants. *ACM Trans. on Computer-Human Interaction (TOCHI)*, 26(3):1–28.

[23] Andow, B., Mahmud, S. Y., Wang, W., Whitaker, J., Enck, W., Reaves, B., Singh, K., and Xie, T. (2019). Policylint: investigating internal privacy policy contradictions on google play. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 585–602.

[24] Andow, B., Mahmud, S. Y., Whitaker, J., Enck, W., Reaves, B., Singh, K., and Egelman, S. (2020). Actions speak louder than words: Entity-sensitive privacy policy and data flow analysis with policheck. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 985–1002.

[25] Andow, B., Nadkarni, A., Bassett, B., Enck, W., and Xie, T. (2016). A study of grayware on google play. In *2016 IEEE Security and Privacy Workshops (SPW)*, pages 224–233. IEEE.

[26] Andrews, G. R. (1991). Paradigms for process interaction in distributed programs. *ACM Comput. Surv.*, 23(1):49–90.

[27] Anthonysamy, P., Edwards, M., Weichel, C., and Rashid, A. (2016). Inferring semantic mapping between policies and code: the clue is in the language. In *International Symposium on Engineering Secure Software and Systems*, pages 233–250. Springer.

[28] Anthonysamy, P., Greenwood, P., and Rashid, A. (2013). Social networking privacy: Understanding the disconnect from policy to controls. *Computer*, 46(6):60–67.

[29] Apthorpe, N., Huang, D. Y., Reisman, D., Narayanan, A., and Feamster, N. (2019). Keeping the smart home private with smart(er) iot traffic shaping. *Proceedings on Privacy Enhancing Technologies*, 2019(3):128–148.

[30] Apthorpe, N., Reisman, D., and Feamster, N. (2017). A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *CoRR*, abs/1705.06805.

[31] Asadollah, S. A., Inam, R., and Hansson, H. (2015). A survey on testing for cyber physical system. In *IFIP International Conference on Testing Software and Systems*, pages 194–207. Springer.

[32] Ava Mutchler (2018). Google assistant app total reaches nearly 2400. https://voicebot.ai/2018/01/24/google-assistant-app-total-reaches-nearly-2400-thats-not-real. [Online; last accessed 22-December-2018].

[33] Avoine, G., Bingol, M. A., Boureanu, I., capkun, S., Hancke, G., Kardas, S., Kim, C. H., Lauradoux, C., Martin, B., Munilla, J., Peinado, A., Rasmussen, K. B., Singelee, D., Tchamkerten, A., Trujillo-Rasua, R., and Vaudenay, S. (2018). Security of distance-bounding: A survey. *ACM Comput. Surv.*, 51(5):94:1–94:33.

[34] Baarslag, T., Alan, A. T., Gomer, R. C., Liccardi, I., Marreiros, H., Gerding, E., et al. (2016). Negotiation as an interaction mechanism for deciding app permissions. In *Proc. of CHI Extended Abstracts*, pages 2012–2019.

[35] Bartel, A., Klein, J., Le Traon, Y., and Monperrus, M. (2012). Automatically securing permission-based software by reducing the attack surface: an application to android. In *2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pages 274–277.

[36] BBC (2020). Amazon alexa security bug allowed access to voice history. https://www.bbc.co.uk/news/technology-53770778. [Online; last accessed 29-September-2021].

[37] Brierley, C., Arief, B., Barnes, D., and Hernandez-Castro, J. (2021). Industrialising blackmail: Privacy invasion based iot ransomware. In Tuveri, N., Michalas, A., and Brumley, B. B., editors, *Secure IT Systems*, pages 72–92, Cham. Springer International Publishing.

[38] Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. (2016). Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 513–530, Austin, TX. USENIX Association.

[39] Carlini, N. and Wagner, D. A. (2016). Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644.

[40] Carlini, N. and Wagner, D. A. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 1–7.

[41] Chalhoub, G. and Flechais, I. (2020). "alexa, are you spying on me?": Exploring the effect of user experience on the security and privacy of smart speaker users. In Moallem, A., editor, *HCI for Cybersecurity, Privacy and Trust*, pages 305–325, Cham. Springer International Publishing.

[42] Chen, K., Wang, P., Lee, Y., Wang, X., Zhang, N., Huang, H., Zou, W., and Liu, P. (2015). Finding unknown malice in 10 seconds: Mass vetting for new threats at the google-play scale. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 659–674.

[43] Chen, S., Ren, K., Piao, S., Wang, C., Wang, Q., Weng, J., Su, L., and Mohaisen, A. (2017). You can hear but you cannot steal:defending against voice impersonation attacks on smartphones. *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*.

[44] Cheng, L., Wilson, C., Liao, S., Young, J., Dong, D., and Hu, H. (2020). Dangerous skills got certified: Measuring the trustworthiness of skill certification in voice personal assistant platforms. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, page In press.

[45] Chowdhury, P. D., Hallett, J., Patnaik, N., Tahaei, M., and Rashid, A. (2021). Developers are neither enemies nor users: They are collaborators. In *2021 IEEE Secure Development Conference (SecDev)*, pages 47–55.

[46] Chung, H. and Lee, S. (2018). Intelligent virtual assistant knows your life. *CoRR*, abs/1803.00466.

[47] Chung, H., Park, J., and Lee, S. (2017). Digital forensic approaches for amazon alexa ecosystem. *Digital Investigation*, 22:S15 to S25.

[48] Ciholas, P., Lennie, A., Sadigova, P., and Such, J. (2019). The security of smart buildings: a systematic literature review. *arXiv preprint arXiv:1901.05837*.

[49] Ciholas, P. and Such, J. (2016). Composite vulnerabilities in cyber physical systems. *Security and Resilience of Cyber–Physical Infrastructures*, page 4.

[50] Cisse, M., Adi, Y., Neverova, N., and Keshet, J. (2017). Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6980–6990.

[51] Commission., F. T. (2013). Children's online privacy protection rule; final rule.

[52] Confessore, N. (2021). Cambridge Analytica and Facebook: The Scandal and the Fallout So Far. https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html. [Online; last accessed 18-August-2021].

[53] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale.

[54] Corchado, E. and Herrero, Á. (2011). Neural visualization of network traffic data for intrusion detection. *Applied Soft Computing*, 11(2):2042–2056.

[55] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

[56] Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., Doumouro, C., Gisselbrecht, T., Caltagirone, F., Lavril, T., Primet, M., and Dureau, J. (2018). Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

[57] Crabtree, A., Lodge, T., Colley, J., Greenhalgh, C., Glover, K., Haddadi, H., Amar, Y., Mortier, R., Li, Q., and Moore, J. e. a. (2018). Building accountability into the internet of things: the iot databox model. *Journal of Reliable Intelligent Environments*, 4(1):39–55.

[58] Criado, N., Argente, E., and Botti, V. (2011). Open issues for normative multi-agent systems. *AI communications*, 24(3):233–264.

[59] Criado, N. and Such, J. (2015). Implicit contextual integrity in online social networks. *Information Sciences*, 325:48–69.

[60] Criado, N. and Such, J. (2016). Selective norm monitoring. In *IJCAI*, pages 208–214.

[61] Cufoglu, A. (2014). User profiling-a short review. *International Journal of Computer Applications (0975 8887)*, 108(3).

[62] Denning, T., Kohno, T., and Levy, H. M. (2013). Computer security and the modern home. *Commun. ACM*, 56(1):94–103.

[63] Developer, G. (2019). Developer Preview of Local Home SDK. https://developers.googleblog.com/2019/07/developer-preview-of-local-home-sdk.html. [Online; last accessed 7-January-2020].

[64] Dou, Z., Khalil, I., Khreishah, A., Al-Fuqaha, A., and Guizani, M. (2017). Systematization of knowledge (sok): A systematic review of software-based web phishing detection. *IEEE Communications Surveys & Tutorials*, 19(4):2797–2819.

[65] Douceur, J. R. (2002). The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer.

[66] Dubois, D. J., Kolcun, R., Mandalari, A. M., Paracha, M. T., Choffnes, D., and Haddadi, H. (2020). When speakers are all ears: Characterizing misactivations of iot smart speakers. *Proceedings on Privacy Enhancing Technologies*, 2020(4):255–276.

[67] Dyer, K. P., Coull, S. E., Ristenpart, T., and Shrimpton, T. (2012). Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *2012 IEEE Symposium on Security and Privacy*, pages 332–346.

[68] Easwara Moorthy, A. and Vu, L. (2015). Privacy concerns for use of voice activated personal assistant in the public space. *International Journal of Human Computer Interaction*, 31(4):307 to 335.

[69] Felke-Morris, T. (2011). *Web development and design foundations with XHTML*. Pearson.

[70] Felt, A. P., Greenwood, K., and Wagner, D. (2010). The effectiveness of install-time permission systems for third-party applications. Technical Report UCB/EECS-2010-143, EECS Department, University of California, Berkeley.

[71] Feng, H., Fawaz, K., and Shin, K. G. (2017). Continuous authentication for voice assistants. *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking - MobiCom '17*.

[72] Fernandes, E., Jung, J., and Prakash, A. (2016a). Security analysis of emerging smart home applications. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 636–654.

[73] Fernandes, E., Paupore, J., Rahmati, A., Simionato, D., Conti, M., and Prakash, A. (2016b). Flowfence: Practical data protection for emerging iot application frameworks. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 531–548, Austin, TX. USENIX Association.

[74] Fette, I., Sadeh, N., and Tomasic, A. (2007). Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web*, pages 649–656.

[75] Fogues, R., Murukannaiah, P. K., Such, J., and Singh, M. P. (2017). Sharing policies in multiuser privacy scenarios: Incorporating context, preferences, and arguments in decision making. *ACM TOCHI*, 24(1):5.

[76] Freed, D., Palmer, J., Minchala, D., Levy, K., Ristenpart, T., and Dell, N. (2018). A stalker's paradise: How intimate partner abusers exploit technology. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 667. ACM.

[77] Fruchter, N. and Liccardi, I. (2018). Consumer attitudes towards privacy and security in home assistants. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI EA '18, page 1–6, New York, NY, USA. Association for Computing Machinery.

[78] Gamba, J., Rashed, M., Razaghpanah, A., Tapiador, J., and Vallina-Rodriguez, N. (2020). An analysis of pre-installed android software. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1039–1055. IEEE.

[79] Garimella, S., Mandal, A., Strom, N., Hoffmeister, B., Matsoukas, S., and Parthasarathi, S. H. K. (2015). Robust i-vector based adaptation of dnn acoustic model for speech recognition. In *INTERSPEECH*.

[80] GAVRILA, V., BAJENARU, L., and DOBRE, C. (2019). Modern single page application architecture: A case study. *Studies in Informatics and Control*, 28(2).

[81] Gong, Y. and Poellabauer, C. (2017). Crafting adversarial examples for speech paralinguistics applications. *CoRR*, abs/1711.03280.

[82] Goodfellow, I., McDaniel, P., and Papernot, N. (2018). Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7):56–66.

[83] Google (2017). Set up multiple users for your speaker or smart display. https://support.google.com/assistant/answer/9071681. [Online; last accessed 20-February-2018].

[84] Google (2018a). Actions on Google. https://developers.google.com/actions/samples/. [Online; last accessed 29-December-2018].

[85] Google (2018b). Invocation and Discovery. https://developers.google.com/actions/sdk/invocation-and-discovery. [Online; last accessed 17-December-2018].

[86] Google (2021). Lighthouse. [Online; last accessed 21-October-2021].

[87] Gosselin, D. (2011). *Javascript*. Course Technology.

[88] Guo, Z., Lin, Z., Li, P., and Chen, K. (2020). Skillexplorer: Understanding the behavior of skills in large scale. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2649–2666. USENIX Association.

[89] Haack, W., Severance, M., Wallace, M., and Wohlwend, J. (2017). Security analysis of amazon echo.

[90] Hallett, J., Patnaik, N., Shreeve, B., and Rashid, A. (2021). "do this! do that!, and nothing will happen" do specifications lead to securely stored passwords? In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 486–498.

[91] Hannay, J. E., Sjoberg, D. I., and Dyba, T. (2007). A systematic review of theory use in software engineering experiments. *IEEE Transactions on Software Engineering*, 33(2):87–107.

[92] Harkous, H., Fawaz, K., Lebret, R., Schaub, F., Shin, K. G., and Aberer, K. (2018). Polisis: Automated analysis and presentation of privacy policies using deep learning. In *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, page 531–548, USA. USENIX Association.

[93] He, W., Golla, M., Padhi, R., Ofek, J., Durmuth, M., Fernandes, E., and Ur, B. (2018). Rethinking access control and authentication for the home internet of things (iot). In *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, pages 255–272, Berkeley, CA, USA. USENIX Association.

[94] Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.

[95] Horton, H. (2018). Amazon Alexa recorded owner's conversation and sent to 'random' contact, couple complains. www.telegraph.co.uk/news/2018/05/25/amazon-alexa-recorded-owners-conversation-sent-random-contact/. [Online; last accessed 17-December-2018].

[96] Hoy, M. B. (2018). Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, 37(1):81–88.

[97] Hu, H., Yang, L., Lin, S., and Wang, G. (2020a). Security vetting process of smart-home assistant applications: A first look and case studies. *arXiv*.

[98] Hu, X., Suarez-Tangil, G., and Sastry, N. (2020b). Multi-country study of third party trackers from real browser histories. In *IEEE European Symposium on Security and Privacy*.

[99] ICO (2021). https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/the-right-to-be-informed/what-methods-can-we-use-to-provide-privacy-information/.

[100] Instruments, T. (2013). AN-1973 Benefits and Challenges of High-Frequency Regulators. http://www.ti.com/lit/an/snva399a/snva399a.pdf. [Online; last accessed 17-December-2018].

[101] internetsociety (2019). https://www.internetsociety.org/wp-content/uploads/2019/09/IoT-Privacy-Brief_20190912_Final-EN.pdf.

[102] Iordanou, C., Smaragdakis, G., Poese, I., and Laoutaris, N. (2018). Tracing Cross Border Web Tracking. In *Proceedings of ACM IMC 2018*, Boston, MA.

[103] Jeffrey, D. N. (2021). Supreme court vacates linkedin-hiq scraping decision, remands to ninth circuit for another look.

[104] Jia, Y., Chen, Q. A., Wang, S., Rahmati, A., Fernandes, E., Mao, Z., and Prakash, A. (2017). Contexiot: Towards providing contextual integrity to appified iot platforms. In *ndss.2017*.

[105] Jones, B. (2017). Mattel Cancels AI-Powered Smart Speaker for Kids Over Privacy Concerns. https://futurism.com/mattel-cancels-ai-powered-smart-speaker-for-kids-over-privacy-concerns. [Online; last accessed 21-October-2021].

[106] Kadianakis, G., Roberts, C. V., Roberts, L. M., and Winter, P. (2017). Anomalous keys in tor relays. *CoRR*, abs/1704.00792.

[107] Kafali, Ö., Ajmeri, N., and Singh, M. P. (2016). Revani: Revising and verifying normative specifications for privacy. *IEEE Intelligent Systems*, 31(5):8–15.

[108] Kafali, Ö., Jones, J., Petruso, M., Williams, L., and Singh, M. P. (2017). How good is a security policy against real breaches? a hipaa case study. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 530–540. IEEE.

[109] KAMM, C. (1995). User interfaces for voice applications. *Colloquium Paper*, 92:10031–10037.

[110] Kathryn, M. (2018). Tell the smart house to mind its own business!: Maintaining privacy and security in the era of smart devices. *Fordham Law Review*, 86.

[111] Kay, M. and Terry, M. (2010). Textured agreements: Re-envisioning electronic consent. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS '10, New York, NY, USA. Association for Computing Machinery.

[112] Kelley, P. G., Bresee, J., Cranor, L. F., and Reeder, R. W. (2009). A "nutrition label" for privacy. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, New York, NY, USA. Association for Computing Machinery.

[113] Kelly, H. (2017). Apple's HomePod is coming. Here's what you need to know about smart speakers. http://money.cnn.com/2017/06/08/technology/gadgets/apple-homepod-smart-speaker-faq/index.html. [Online; last accessed 21-December-2018].

[114] Kelly, M. and Statt, N. (2019). Amazon confirms it holds on to Alexa data even if you delete audio files. https://www.theverge.com/2019/7/3/20681423/amazon-alexa-echo-chris-coons-data-transcripts-recording-privacy. [Online; last accessed 21-December-2019].

[115] Kepuska, V. and Bohouta, G. (2018). Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 99–103.

[116] Kim, Y., Kim, D., Kim, J., and Sarikaya, R. (2018). A scalable neural shortlisting-reranking approach for large-scale domain classification in natural language understanding. *CoRR*, abs/1804.08064.

[117] Kinsella, B. (2018a). Alexa Skill Store for France is a Fast Growing Land of Opportunity. https://voicebot.ai/2018/11/03/the-alexa-skill-store-for-france-is-a-fast-growing-land-of-opportunity/. [Online; last accessed 22-December-2018].

[118] Kinsella, B. (2018b). Amazon Introduces Skill Connections so Alexa Skills Can Work Together. https://voicebot.ai/2018/10/04/amazon-introduces-skill-connections-so-alexa-skills-can/. [Online; last accessed 24-December-2018].

[119] Kinsella, B. (2019). Google assistant actions total 4,253 in january 2019. https://voicebot.ai/2019/02/15/google-assistant-actions-total-4253-in-january-2019-up-2-5x-in-past-year-but-7-5-the-total-number-alexa-skills-in-u-s/. [Online; last accessed June-2021].

[120] Kinsella, B. (2020a). Alexa becomes a chatbot — you can now talk to alexa by typing.

[121] Kinsella, B. (2020b). Amazon Launches New Alexa App Updates and Takes Its Mobile Strategy Another Step Forward. https://perma.cc/CYJ6-9JSH.

[122] Kinsella, B. (2020c). Nearly 90 million u.s. adults have smart speakers. https://voicebot.ai/2020/04/28/nearly-90-million-u-s-adults-have-smart-speakers. [Online; last accessed May-2020].

[123] Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering a systematic literature review. *Information and Software Technology*, 51(1):7–15.

[124] Knijnenburg, B. and Cherry, D. (2016). Comics as a medium for privacy notices. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, Denver, CO. USENIX Association.

[125] Knowles, W., Such, J., Gouglidis, A., Misra, G., and Rashid, A. (2015). Assurance techniques for industrial control systems (ics). In *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, pages 101–112. ACM.

[126] Kotzias, P., Caballero, J., and Bilge, L. (2021). How did that get in my phone? unwanted app distribution on android devices. In *IEEE Symposium on Security and Privacy (SP)*.

[127] Kumar, D., Paccagnella, R., Murley, P., Hennenfent, E., Mason, J., Bates, A., and Bailey, M. (2018). Skill squatting attacks on amazon alexa. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 33–47, Baltimore, MD. USENIX Association.

[128] Lai, A. (2018). Sneaky Kid Orders $350 Worth of Toys on Her Mom's Amazon Account. https://mom.me/news/271144-sneaky-kid-orders-350-worth-toys-her-moms-amazon-account/. [Online; last accessed 17-December-2018].

[129] Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

[130] Lau, J., Zimmerman, B., and Schaub, F. (2018). Alexa, are you listening?: Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW):102:1–102:31.

[131] Lavrentyeva, G., Novoselov, S., Malykh, E., Kozlov, A., Kudashev, O., and Shchemelinin, V. (2017). Audio-replay attack detection countermeasures. *CoRR*, abs/1705.08858.

[132] Lee, K.-M. and Nass, C. (2005). Social-psychological origins of feelings of presence: Creating social presence with machine generated voices. *Media Psychology*, 7(1):31–45.

[133] Lei, X., Tu, G.-H., Liu, A. X., Li, C.-Y., and Xie, T. (2018). The insecurity of home digital voice assistants - vulnerabilities, attacks and countermeasures. In *2018 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9.

[134] Lentzsch, C., Shah, S. J., Andow, B., Degeling, M., Das, A., and Enck, W. (2021). Hey alexa, is this skill safe?: Taking a closer look at the alexa skill ecosystem. In *Network and Distributed Systems Security (NDSS) Symposium*.

[135] Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. In *Doklady Akademii Nauk SSSR*, pages 845–848.

[136] Liao, S., Wilson, C., Cheng, L., Hu, H., and Deng, H. (2020). Measuring the effectiveness of privacy policies for voice assistant applications.

[137] Liu, J., Zhang, C., and Fang, Y. (2018). Epic: A differential privacy framework to defend smart homes against internet traffic analysis. *IEEE Internet of Things Journal*, 5(2):1206–1217.

[138] Londhe, N. D., Ahirwal, M. K., and Lodha, P. (2016). Machine learning paradigms for speech recognition of an indian dialect. *2016 International Conference on Communication and Signal Processing (ICCSP).*

[139] Luger, E. and Sellen, A. (2016). "like having a really bad pa": The gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5286–5297, New York, NY, USA. ACM.

[140] Madaan, N., Ahad, M. A., and Sastry, S. M. (2018). Data integration in iot ecosystem: Information linkage as a privacy threat. *Computer Law and Security Review*, 34(1):125–133.

[141] Madjarov, G., Kocev, D., Gjorgjevikj, D., and Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern recognition*, 45(9):3084–3104.

[142] Major, D., Huang, D. Y., Chetty, M., and Feamster, N. (2021). Alexa, who am i speaking to?: Understanding users' ability to identify third-party apps on amazon alexa. *ACM Transactions on Internet Technology (TOIT)*, 22(1):1–22.

[143] Malik, K. M., Malik, H., and Baumann, R. (2019). Towards vulnerability analysis of voice-driven interfaces and countermeasures for replay attacks. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 523–528.

[144] Malkin, N., Deatrick, J., Tong, A., Wijesekera, P., Egelman, S., and Wagner, D. (2019). Privacy attitudes of smart speaker users. *Proceedings on Privacy Enhancing Technologies*, 2019(4):250–271.

[145] Mandal, M. W. S. P. M. S. J. G. R. T. S. N. P. V. B. H. A. (2018). Monophone-based background modeling for two-stage on-device wake word detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5494–5498. IEEE.

[146] Martin, R. C. (2000). Design Principles and Design Patterns. https://fi.ort.edu.uy/innovaportal/file/2032/1/design_principles.pdf. [Online; last accessed 21-October-2021].

[147] Martin, T. (2018). 12 reasons to use Alexa in the kitchen. https://www.cnet.com/how-to/how-to-use-alexa-in-the-kitchen/. [Online; last accessed 17-December-2018].

[148] Matthews, T., Liao, K., Turner, A., Berkovich, M., Reeder, R., and Consolvo, S. (2016). She'll just grab any device that's closer: A study of everyday device & account sharing in households. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5921–5932. ACM.

[149] Matthews, T., O'Leary, K., Turner, A., Sleeper, M., Woelfer, J. P., Shelton, M., Manthorne, C., Churchill, E. F., and Consolvo, S. (2017). Security and privacy experiences and practices of survivors of intimate partner abuse. *IEEE Security Privacy*, 15(5):76–81.

[150] Mayer, J. R. and Mitchell, J. C. (2012). Third-party web tracking: Policy and technology. In *2012 IEEE Symposium on Security and Privacy*, pages 413–427. IEEE.

[151] Memon, A. M. and Anwar, A. (2015). Colluding apps: Tomorrow's mobile malware threat. *IEEE Security & Privacy*, 13(6):77–81.

[152] Meng, N., Keküllüoğlu, D., and Vaniea, K. (2021). Owning and sharing: Privacy perceptions of smart speaker users. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–29.

[153] Misra, G. and Such, J. (2017a). Pacman: Personal agent for access control in social media. *IEEE Internet Computing*, 21(6):18–26.

[154] Misra, G. and Such, J. (2017b). React: Recommending access control decisions to social media users. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 421–426.

[155] Misra, G., Such, J., and Gill, L. (2017). A privacy assessment of social media aggregators. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 561–568. ACM.

[156] Mitrokotsa, A., Dimitrakakis, C., Peris-Lopez, P., and Hernandez-Castro, J. C. (2010). Reid et al.'s distance bounding protocol and mafia fraud attacks over noisy channels. *IEEE Communications Letters*, 14(2):121–123.

[157] Modi, C., Patel, D., Borisaniya, B., Patel, A., and Rajarajan, M. (2012). A survey on security issues and solutions at different layers of cloud computing. *The Journal of Supercomputing*, 63(2):561–592.

[158] Molich, R. and Nielsen, J. (1990). Improving a human-computer dialogue. *Commun. ACM*, 33(3):338–348.

[159] Mosner, L., Wu, M., Raju, A., Krishnan Parthasarathi, S. H., Kumatani, K., Sundaram, S., Maas, R., and Hoffmeister, B. (2019). Improving noise robustness of automatic speech recognition via parallel data and teacher-student learning. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[160] Murdoch, S. J. and Zielinski, P. (2007). Sampled traffic analysis by internet-exchange-level adversaries. In *Proceedings of the 7th International Conference on Privacy Enhancing Technologies*, PET'07, pages 167–183, Berlin, Heidelberg. Springer-Verlag.

[161] Mylonas, A., Theoharidou, M., and Gritzalis, D. (2014). Assessing privacy risks in android: A user-centric approach. In Bauer, T., Großmann, J., Seehusen, F., Stølen, K., and Wendland, M.-F., editors, *Risk Assessment and Risk-Driven Testing*, pages 21–37, Cham. Springer International Publishing.

[162] Naiakshina, A., Danilova, A., Gerlitz, E., von Zezschwitz, E., and Smith, M. (2019). "if you want, i can store the encrypted password": A password-storage field study with freelance developers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

[163] Naik, C., Gupta, A., Ge, H., Lambert, M., and Sarikaya, R. (2018). Contextual slot carryover for disparate schemas. In *Proc. Interspeech 2018*, pages 596–600.

[164] Nass, C., Moon, Y., and Carney, P. (1999). Are people polite to computers? responses to computer-based interviewing systems1. *Journal of Applied Social Psychology*, 29(5):1093–1109.

[165] Natatsuka, A., Iijima, R., Watanabe, T., Akiyama, M., Sakai, T., and Mori, T. (2019). Poster: A first look at the privacy risks of voice assistant apps. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 2633–2635, New York, NY, USA. Association for Computing Machinery.

[166] Newman, L. H. (2018). Millions of Streaming Devices Are Vulnerable to a Retro Web Attack. https://www.wired.com/story/chromecast-roku-sonos-dns-rebinding-vulnerability/. [Online; last accessed 21-April-2020].

[167] Ni, X., Yang, Z., Bai, X., Champion, A. C., and Xuan, D. (2009). Diffuser: Differentiated user access control on smartphones. In *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, pages 1012–1017.

[168] Nissenbaum, H. (2004). Privacy as contextual integrity. *Washington Law Review*, 79:119.

[169] Olejnik, K., Dacosta, I., Machado, J. S., Huguenin, K., Khan, M. E., and Hubaux, J.-P. (2017). Smarper: Context-aware and automatic runtime-permissions for mobile devices. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 1058–1076. IEEE.

[170] OVUM (2017). Virtual digital assistants to overtake world population by 2021.

[171] Pagliardini, M., Gupta, P., and Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

[172] Papayiannis, C., Amoh, J., Rozgic, V., Sundaram, S., and Wang, C. (2018). Detecting media sound presence in acoustic scenes. In *Proc. Interspeech 2018*, pages 1363–1367.

[173] Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. (2018). Towards the science of security and privacy in machine learning. In *3rd IEEE European Symposium on Security and Privacy.*

[174] Papernot, N., McDaniel, P. D., and Goodfellow, I. J. (2016). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277.

[175] Parikh, A. P., Tackstrom, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In *Proceedings of EMNLP*.

[176] Park, H., Basaran, C., Park, T., and Son, S. (2014). Energy-efficient privacy protection for smart home environments using behavioral semantics. *Sensors*, 14(9):16235–16257.

[177] Peachman, R. R. (2017). Mattel Pulls Aristotle Children's Device After Privacy Concerns. https://www.nytimes.com/2017/10/05/well/family/mattel-aristotle-privacy.html. [Online; last accessed 21-October-2021].

[178] Perera, C., Wakenshaw, S. Y. L., Baarslag, T., Haddadi, H., Bandara, A. K., Mortier, R., Crabtree, A., Ng, I. C. L., McAuley, D., and Crowcroft, J. (2017). Valorising the iot databox: creating value for everyone. *Transactions on Emerging Telecommunications Technologies*, 28(1):e3125. e3125 ett.3125.

[179] Perez, A. J., Zeadally, S., and Cochran, J. (2018). A review and an empirical analysis of privacy policy and notices for consumer internet of things. *Security and Privacy*, 1(3).

[180] Perez, S. (2016). Amazon Alexa now has over 1,000 Skills, up from 135 in January. https://techcrunch.com/2016/06/03/amazon-alexa-now-has-over-1000-skills-up-from-135. [Online; last accessed May-2020].

[181] Picchi, A. (2019). Amazon workers are listening to what you tell Alexa. https://www.cbsnews.com/news/amazon-workers-are-listening-to-what-you-tell-alexa/. [Online; last accessed 21-February-2020].

[182] Pinyol, I. and Sabater-Mir, J. (2013). Computational trust and reputation models for open multi-agent systems: a review. *Artif Intell Rev*, 40(1):1–25.

[183] Ponticello, A., Fassl, M., and Krombholz, K. (2021). Exploring authentication for security-sensitive tasks on smart home voice assistants. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*.

[184] Prandini, M. and Ramilli, M. (2010). Towards a practical and effective security testing methodology. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pages 320–325. IEEE.

[185] Priyanka, C. and Rajendra, S. (2016). Security attacks on cloud computing with possible solution.

[186] Purington, A., Taft, J. G., Sannon, S., Bazarova, N. N., and Taylor, S. H. (2017). "alexa is my new bff": Social roles, user satisfaction, and personification of the amazon echo. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, pages 2853–2859, New York, NY, USA. ACM.

[187] Ramokapane, K. M., Rashid, A., and Such, J. (2016). Assured deletion in the cloud: requirements, challenges and future directions. In *Proceedings of the 2016 ACM on Cloud Computing Security Workshop*, pages 97–108.

[188] Ramokapane, K. M., Rashid, A., and Such, J. (2017). “i feel stupid i can’t delete...”: A study of users’ cloud deletion practices and coping strategies. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 241–256.

[189] Reid, T. (2018). Everything Alexa learned in 2018. https://blog.aboutamazon.com/devices/everything-alexa-learned-in-2018. [Online; last accessed 4-January-2019].

[190] Ren, J., Dubois, D. J., Choffnes, D., Mandalari, A. M., Kolcun, R., and Haddadi, H. (2019). Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In *Proceedings of the Internet Measurement Conference*, IMC ’19, page 267–279, New York, NY, USA. Association for Computing Machinery.

[191] Roberts, M. L. (2019). Are Your Voice Assistants Always Listening? The simplistic answer is "Yes"... http://www.capecodtoday.com/article/2019/08/11/248280-Are-Your-Voice-Assistants-Always-Listening. [Online; last accessed 21-February-2020].

[192] Rodehorst, M. (2019). Why Alexa Won’t Wake Up When She Hears Her Name in Amazon’s Super Bowl Ad. http://web.archive.org/web/20190211063816/https://developer.amazon.com/blogs/alexa/post/37857f29-dd82-4cf4-9ebd-6ebe632f74d3/why-alexa-won-t-wake-up-when-she-hears-her-name-in-amazon-s-super-bowl-ad. [Online; last accessed 21-March-2020].

[193] Roman, R., Lopez, J., and Gritzalis, S. (2018). Evolution and trends in the security of the internet of things. *IEEE Computer*, 51:16–25.

[194] Roman, R., Rios, R., Onieva, J. A., and Lopez, J. (In Press). Immune system for the internet of things using edge technologies. *IEEE Internet of Things Journal*.

[195] Ronen, E., Shamir, A., Weingarten, A., and Flynn, C. O. (2018). Iot goes nuclear: Creating a zigbee chain reaction. *IEEE Security Privacy*, 16(1):54 to 62.

[196] Roy, N., Shen, S., Hassanieh, H., and Choudhury, R. R. (2018). Inaudible voice commands: The long-range attack and defense. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 547–560, Renton, WA. USENIX Association.

[197] Ruan, S., Wobbrock, J. O., Liou, K., Ng, A., and Landay, J. A. (2018). Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–23.

[198] Saidi, S. J., Mandalari, A. M., Haddadi, H., Dubois, D. J., Choffnes, D., Smaragdakis, G., and Feldmann, A. (2021). Detecting consumer iot devices through the lens of an isp. In *Proceedings of the Applied Networking Research Workshop*, ANRW ’21, page 36–38, New York, NY, USA. Association for Computing Machinery.

[199] Sarma, B. P., Li, N., Gates, C., Potharaju, R., Nita-Rotaru, C., and Molloy, I. (2012). *Android Permissions: A Perspective Combining Risks and Benefits*, page 13–22. Association for Computing Machinery, New York, NY, USA.

[200] Schaub, F., Balebako, R., and Cranor, L. F. (2017). Designing effective privacy notices and controls. *IEEE Internet Computing*, 21(3):70–77.

[201] Schonherr, L., Kohls, K., Zeiler, S., Holz, T., and Kolossa, D. (2018). Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *CoRR*, abs/1808.05665.

[202] School, C. L. (2021). Van buren v. united states 940 f. 3d 1192.

[203] Sciuto, A., Saini, A., Forlizzi, J., and Hong, J. I. (2018). "Hey Alexa, What's Up?" a mixed-methods studies of in-home conversational agent usage. In *Procs of the Designing Interactive Systems Conference*, pages 857–868.

[204] Scoccia, G. L., Peruma, A., Pujols, V., Malavolta, I., and Krutz, D. E. (2019). Permission issues in open-source android apps: An exploratory study. In *2019 19th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 238–249.

[205] Selenium (2020). Selenium webdriver. https://www.selenium.dev/. [Online; last accessed 15-October-2020].

[206] Sharevski, F., Jachim, P., Treebridge, P., Li, A., Babin, A., and Adadevoh, C. (2021). Meet malexa, alexa's malicious twin: Malware-induced misperception through intelligent voice assistants. *International Journal of Human-Computer Studies*, 149:102604.

[207] Shayegh, P. and Ghanavati, S. (2017). Toward an approach to privacy notices in iot. *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*.

[208] Shezan, F. H., Hu, H., Wang, J., Wang, G., and Tian, Y. (2020). Read between the lines: An empirical measurement of sensitive applications of voice personal assistant systems. In *Proceedings of The Web Conference 2020*, pages 1006–1017.

[209] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

[210] Singleton, M. (2017). Alexa can now set reminders for you. https://www.theverge.com/circuitbreaker/2017/6/1/15724474/alexa-echo-amazon-reminders-named-timers. [Online; last accessed 21-December-2018].

[211] Solove, D. (2006). A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3):477–560.

[212] SRLabs (2019). Smart Spies: Alexa and Google Home expose users to vishing and eavesdropping. https://srlabs.de/bites/smart-spies/. [Online; last accessed 21-February-2020].

[213] Stanfordnlp (2021). https://stanfordnlp.github.io/CoreNLP/. [Online; last accessed 29-May-2021].

[214] Statista (2018). Worldwide intelligent/digital assistant market share in 2017 and 2020, by product. https://www.statista.com/statistics/789633/worldwide-digital-assistant-market-share/. [Online; last accessed 21-December-2018].

[215] Statt, N. (2019). Google defends letting human workers listen to Assistant voice conversations. https://www.theverge.com/2019/7/11/20691021/google-assistant-ai-training-controversy-human-workers-listening-privacy.

[216] Stringhini, G., Kruegel, C., and Vigna, G. (2010). Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference*, pages 1–9.

[217] Suarez-Tangil, G., Tapiador, J. E., Peris-Lopez, P., and Ribagorda, A. (2014). Evolution, detection and analysis of malware in smart devices. *IEEE Communications Surveys & Tutorials*, 16(2):961–987.

[218] Such, J. (2017). Privacy and autonomous systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4761–4767. AAAI Press.

[219] Such, J., Ciholas, P., Rashid, A., Vidler, J., and Seabrook, T. (2019). Basic cyber hygiene: Does it work? *Computer*, 52(4):21–31.

[220] Such, J. and Criado, N. (2016). Resolving multi-party privacy conflicts in social media. *IEEE TKDE*, 28(7):1851–1863.

[221] Such, J. and Criado, N. (2018). Multiparty privacy in social media. *Communications of the ACM*, 61(8):74–81.

[222] Such, J., Criado, N., Vercouter, L., and Rehak, M. (2016a). Intelligent cybersecurity agents. *IEEE Intelligent Systems*, 31(5):3–7.

[223] Such, J., Gouglidis, A., Knowles, W., Gaurav, M., and Awais, R. (2016b). Information assurance techniques: Perceived cost effectiveness. *Computers and Security*, 60:117–133.

[224] Such, J. and Rovatsos, M. (2016). Privacy policy negotiation in social media. *ACM Trans. on Autonomous and Adaptive Systems*, 11(1):4.

[225] Sugawara, T., Cyr, B., Rampazzi, S., Genkin, D., and Fu, K. (2020). Light commands: Laser-Based audio injection attacks on Voice-Controllable systems. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2631–2648. USENIX Association.

[226] Syverson, P. (2009). Why i'm not an entropist. In *In the Proceedings of Security Protocols XVII: 17th International Workshop.*

[227] Syverson, P. (2011). Sleeping dogs lie on a bed of onions but wake when mixed. In *Proceedings of HotPETS 2011.*

[228] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2013). Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

[229] Tabassum, M., Alqhatani, A., Aldossari, M., and Richter Lipford, H. (2018). *Increasing User Attention with a Comic-Based Policy*, page 1–6. Association for Computing Machinery, New York, NY, USA.

[230] Tabassum, M., Kosiński, T., Frik, A., Malkin, N., Wijesekera, P., Egelman, S., and Lipford, H. R. (2019a). Investigating users' preferences and expectations for always-listening voice assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–23.

[231] Tabassum, M., Kosinski, T., and Lipford, H. R. (2019b). "i don't own the data": End user perceptions of smart home device data practices and risks. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, Santa Clara, CA. USENIX Association.

[232] Todisco, M., Delgado, H., and Evans, N. (2017). Constant q cepstral coefficients. *Comput. Speech Lang.*, 45(C):516–535.

[233] Vaidya, T., Zhang, Y., Sherr, M., and Shields, C. (2015). Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, Washington, D.C. USENIX Association.

[234] van der Linden, D., Anthonysamy, P., Nuseibeh, B., Tun, T. T., Petre, M., Levine, M., Towse, J., and Rashid, A. (2020). Schrödinger's security: Opening the box on app developers' security rationale. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 149–160.

[235] Wang, H., Guo, Y., Tang, Z., Bai, G., and Chen, X. (2015). Reevaluating android permission gaps with static and dynamic analysis. In *2015 IEEE Global Communications Conference (GLOBECOM)*.

[236] Wang, X., Hou, X., Rios, R., Hallgren, P., Tippenhauer, N. O., and Ochoa, M. (2018a). Location proximity attacks against mobile targets. In *23rd European Symposium on Research in Computer Security (ESORICS 2018)*, volume 11099 of *LNCS*, pages 373–392, Barcelona. Springer, Springer.

[237] Wang, X., Qin, X., Bokaei Hosseini, M., Slavin, R., Breaux, T. D., and Niu, J. (2018b). Guileak: Tracing privacy policy claims on user input data for android applications. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pages 37–47.

[238] Watch, H. R. (2017). China: Voice biometric collection threatens privacy.

[239] White, R. W. (2018). Skill discovery in virtual assistants. *Communications of the ACM*, 61(11):106–113.

[240] Wolfson, S. (2018). Amazon's Alexa recorded private conversation and sent it to random contact. www.theguardian.com/technology/2018/may/24/amazon-alexa-recorded-conversation. [Online; last accessed 17-December-2018].

[241] Wong, V. (2017). Burger King's New Ad Will Hijack Your Google Home. https://www.cnbc.com/2017/04/12/burger-kings-new-ad-will-hijack-your-google-home.html. [Online; last accessed 25-December-2018].

[242] Wright, C., Coull, S., and Monrose, F. (2009). Traffic morphing: An efficient defense against statistical traffic analysis. In *Proceedings of the Network and Distributed Security Symposium*. IEEE.

[243] Wright, D. and De Hert, P. (2012). Introduction to privacy impact assessment. In *Privacy Impact Assessment*, pages 3–32. Springer.

[244] Xu, H., Durme, B. V., and Murray, K. W. (2021). Bert, mbert, or bibert? A study on contextualized embeddings for neural machine translation. *CoRR*, abs/2109.04588.

[245] Yang, J. (2018). Multilayer adaptation based complex echo cancellation and voice enhancement. *2018 IEEE Int. conference on Acoustics, Speech and Signal Processing (ICASSP).*

[246] Yilmaz, Y., Cetin, O., Arief, B., and Hernandez-Castro, J. (2021). Investigating the impact of ransomware splash screens. *Journal of Information Security and Applications*, 61:102934.

[247] Young, J., Liao, S., Cheng, L., Hu, H., and Deng, H. (2022). SkillDetective: Automated Policy-Violation detection of voice assistant applications in the wild. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA. USENIX Association.

[248] Young, J. D. and Anton, A. I. (2010). A method for identifying software requirements based on policy commitments. In *2010 18th IEEE International Requirements Engineering Conference*, pages 47–56. IEEE.

[249] Zeng, E., Mare, S., and Roesner, F. (2017). End user security and privacy concerns with smart homes. In *Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security*, SOUPS'17, pages 65–80, Berkeley, CA, USA. USENIX Association.

[250] Zeng, E. and Roesner, F. (2019). Understanding and improving security and privacy in multi-user smart homes: a design exploration and in-home user study. In *28th USENIX Security Symposium USENIX Security 19).*

[251] Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., and Xu, W. (2017). Dolphin attack. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17.*

[252] Zhang, N., Mi, X., Feng, X., Wang, X., Tian, Y., and Qian, F. (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)*, pages 1381–1396.

[253] Zhao, S., Yang, W., Wang, D., and Qiu, W. (2012). A new scheme with secure cookie against sslstrip attack. In Wang, F. L., Lei, J., Gong, Z., and Luo, X., editors, *Web Information Systems and Mining*, pages 214–221, Berlin, Heidelberg. Springer Berlin Heidelberg.

[254] Zhao, Y., Haddadi, H., Skillman, S., Enshaeifar, S., and Barnaghi, P. (2020). Privacy-preserving activity and health monitoring on databox. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, EdgeSys '20, page 49–54, New York, NY, USA. Association for Computing Machinery.

[255] Zheng, S., Apthorpe, N., Chetty, M., and Feamster, N. (2018). User perceptions of smart home iot privacy. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW):200:1–200:20.

[256] Zimmeck, S., Story, P., Smullen, D., Ravichander, A., Wang, Z., Reidenberg, J., Russell, N. C., and Sadeh, N. (2019). Maps: Scaling privacy compliance analysis to a million apps. *Proceedings on Privacy Enhancing Technologies*, 2019(3):66–86.

[257] Zolnay, A., Kocharov, D., Schluter, R., and Ney, H. (2007). Using multiple acoustic feature sets for speech recognition. *Speech Communication*, 49(6):514–525.