Don't Enchanted by Siren's Call: An Empirical Study of Privacy Risks in Android Task-executable Voice Assistants

ABSTRACT

As technology continues to evolve, task-executable voice assistants (VAs) have become more popular, enhancing user convenience and expanding device functionality. Task-executable VAs are commonl AI-powered applications that are capable of understanding complex tasks and performing corresponding operations. For example, Amazon Alexa and Google Assistant provide a plenty of daily tasks (e.g., making phone calls, conducting web searches), allowing users to manipulate applications by simple voice commands. Despite their prevalence, there is no work examine the privacy and security risks within the voice assistants in a holistic manner. To fill this research gap, this paper presents the first comprehensive empirical study on privacy and security risks in Android voice assistant applications. We explored the practical capabilities of voice assistants, investigating the privacy collection, security threads and permission disclosure. Our empirical results on the most popular voice assistants reveal that 1) 20 percent of top voice assistant apps lack proper privacy disclosure, in direct violation of legal requirements; 2) Privilege escalation can occur in the context of cross-app interaction, allowing unauthorized access to sensitive data and further exacerbating security risks; 3) third-party voice assistants can exploit the Google Assistant framework to obtain private data through voice commands without the explicit declaration of the corresponding permissions. We discuss the implications of our findings for developers, platform providers, regulators, and users, emphasizing the need for improved privacy protections and security measures within the voice assistant ecosystem.

ACM Reference Format:

. 2024. Don't Enchanted by Siren's Call: An Empirical Study of Privacy Risks in Android Task-executable Voice Assistants. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Voice assistants (VA) constitute an indispensable technological interface for individuals with visual impairments or those unable to manipulate traditional application interfaces, serving as a critical conduit for accessing digital services and information. These intelligent systems represent a significant advancement in human-computer interaction, offering substantial benefits for users with disabilities. By leveraging natural language processing and machine learning algorithms, voice assistants enable voice-driven command and control mechanisms, thereby enhancing the accessibility of digital technologies [37]. The integration of these systems into everyday devices has the potential to significantly ameliorate the digital divide experienced by individuals with diverse accessibility needs, fostering greater independence and societal inclusion.

While the accessibility benefits of VAs are profound, their impact extends far beyond this domain, as evidenced by their widespread adoption and continuous evolution in the field of human-computer interaction. The rapid advancement of artificial intelligence and associated technologies has propelled VAs to the forefront of human-computer interaction paradigms. For instance, Apple's Siri [1], Amazon's Alexa [2], and Google Assistant [3] have revolutionized how users interact with their devices, allowing for hands-free operation of smartphones, smart home devices, and even vehicles. [9, 30]. However, it is crucial to distinguish between traditional VAs and task-executable VAs, the latter being the focus of our study. Voice-based assistants like Alexa, Siri, and Google Assistant have evolved beyond simple chatbot functionalities to become sophisticated, task-oriented platforms integral to daily life. These advanced VAs leverage cutting-edge technologies such as Computer Vision and Natural Language Processing to comprehend and execute complex commands across various domains. Task-executable VAs represent a significant leap forward in functionality. Unlike their predecessors, which were primarily limited to answering queries and performing basic tasks, these modern VAs can interact with multiple applications, control smart home devices, make purchases, and even perform complex multi-step operations. This enhanced capability allows them to seamlessly integrate into users' digital ecosystems, offering a more intuitive and efficient way to interact with technology. However, this increased functionality in task-executable VAs also introduces new security and privacy challenges that warrant careful examination and analysis.

The unpredictability of these devices has been further illustrated by several notable incidents. In 2017, a news segment about a child who had ordered a dollhouse via Alexa caused multiple Echo devices in viewers' homes to attempt the same order. The news anchor's repetition of the phrase "Alexa, order a dollhouse" activated the devices, leading to unintended purchases [32]. Similarly, in 2018, a family in Portland, Oregon, discovered that their Amazon Echo device had not only recorded a private conversation without their knowledge but also sent the recording to a random contact in their address book. Amazon later explained that the device had misinterpreted background conversation as a series of commands, leading to the accidental recording and sharing of the conversation. Such incidents raise serious concerns about unauthorized data access and the potential misuse of personal information [44]. This event highlights the ease with which these assistants can be inadvertently triggered, resulting in unintended actions that have real-world consequences.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] https://www.apple.com/au/siri/
[2] https://www.alexa.com/
[3] https://assistant.google.com/

Breakthroughs AI technologies such as speech recognition and natural language processing further contribute the rise of voice assistants. From the initial simple instruction executor to the current provider of intelligent dialogue and personalized services, voice assistants have undergone significant development. The earliest voice assistants could only execute some simple commands, such as setting alarms, sending text messages, etc. Users need to use specific voice commands and the voice recognition rate is low, resulting in an unsatisfactory experience. With the advancement of speech recognition and natural language processing technology, voice assistants have gradually acquired the ability to conduct intelligent conversations. Users can communicate with voice assistants to obtain information, ask questions, and more. Today's voice assistants are no longer just executors and interlocutors, they can also provide personalized services based on users' preferences and habits. For example, reminding activities based on the user's schedule, recommending suitable music and news, etc. [17, 20, 42] However, their increasing intelligence and functionalities further brings significant security and privacy challenges.

As these assistants gain more access to personal data, the risks associated with AI-induced vulnerabilities become more pronounced. More and more AI voice assistant and IoT items had integrate into our daily life. Recent privacy concerns surrounding voice assistants have highlighted significant gaps in data management practices, particularly involving children's data. From BBC June 1 2023 news, Amazon's $25 million settlement with the U.S. Federal Trade Commission for violating the Children's Online Privacy Protection Act (COPPA) underscores these risks. The case revealed that Amazon retained children's voice recordings and location data despite deletion requests, exposing the disconnect between declared privacy policies and actual practices. [45] Unauthorized data access and misuse are not the only concerns; there is also the potential for these devices to be exploited for malicious purposes. The privacy of voice assistant gradually become a really serious problem that we need to concern [1, 6, 26].

Since voice assistants operate as black boxes, the data privacy declaration of voice assistants is critical, and it is the only way users can make sense of the privacy usage of voice assistants. [16, 18, 19, 21, 40] Therefore a detailed and easy to understand privacy statement is a very important way to let the user know what permissions the voice assistant is using. As noted in this paper, while having a privacy statement is no longer a significant issue for current voice assistants, a new concern has emerged: some voice assistants show inconsistencies in their privacy statements across different platforms, or there are discrepancies between the declared privacy practices and actual usage. Moreover, by examining the features of popular voice assistants, we can gain insights into the differences and preferences in privacy declarations from both the market and developers.

Numerous previous studies have analyzed privacy policies from various perspectives. However, most existing compliance analyses primarily focus on data usage, whether it has been properly declared, and what data may have been collected.[1–3, 6, 26, 31] To evaluate current privacy declaration and their actual use in the voice assistant, we investigate the following research questions:
RQ1: Privacy disclosure inconsistency between settings, usage, and declaration. (Section 3)
RQ2: Privacy misdisclosure in mega applications. (Section 4)
RQ3: The fail of maginot line: privilege escalation through inter-application interactions. (Section 5)
RQ4: The abuse of google system application. (Section 6)

We begin by examining the current landscape of voice assistants in the application market, focusing on their capabilities and the extent of permissions they require. This involves analyzing how these devices operate, what data they collect, and the potential risks associated with their use. We then investigate the risks of privacy leakage and issues related to the over- and under-disclosure of permissions in specific voice assistants. This includes a detailed look at how these devices handle user data and the implications of their data management practices. We also consider the legal and regulatory frameworks that govern the use of voice assistants, assessing whether current policies are adequate to address the emerging threats. Finally, we present case studies that delve into privacy factors, illustrating the consequences of inadequate privacy disclosures and the reliance on third-party services. These case studies provide concrete examples of how voice assistants can inadvertently compromise user privacy and highlight the need for more transparent and comprehensive privacy practices.

Contribution. This study presents a comprehensive analysis of privacy and security risks in Android task-executable voice assistants, uncovering critical vulnerabilities and compliance issues. Our findings provide actionable insights for users, developers, and regulators to enhance the safety and reliability of voice-based interfaces. We make the following contributions:

We reveal that 60% of analyzed voice assistant apps fail to meet expected privacy and security standards, with an average of 3.5 undeclared permissions per app, highlighting a significant compliance gap in the industry. We demonstrate how these privacy and security shortcomings can negatively impact user experience, potentially exposing users to unauthorized data access, privacy breaches, and security vulnerabilities when using task-executable voice assistants. We outline critical areas for future work, emphasizing the need for comprehensive automated testing frameworks specifically designed for task-executable voice assistants to address the complex vulnerabilities and privacy issues uncovered in our study.

2 STATUS QUO OF ANDROID TASK-EXECUTABLE VOICE ASSISTANTS

In this section, we briefly introduce the market of Android task-executable voice assistants, capabilities, and characteristics. We manually collected and identified 10 mainstream task-executable voice assistants. We analysed various characteristics and specifically the range of possible permission uses in voice assistants.

2.1 Collecting Task-executable Voice Assistants from Google Play

Task-executable voice assistants are commonly AI-powered applications that are capable of understanding complex tasks and performing corresponding operations on mobile phones or connected IoT devices. For example, Amazon Alexa can invoke the TED Talks app on the phone by the voice command "Alexa, ask TED

Table 1: Examples of voice assistant applications that are excluded based on selection criteria, *i.e.*, not task-executable.

Type	App Name	Package Name	Developer	Downloads
Task-executable Voice Assistants	Google Assistant	com.google.android.apps.googleassistant	Google	1B+
	Voice Access	com.google.android.apps.accessibility.voiceaccess	Google	100M+
	Amazon Alexa	com.amazon.dee.app	Amazon Mobile	100M+
	Ultimate Alexa Voice Assistant	com.customsolutions.android.alexa	Custom Solutions	5M+
	Voice Search	ru.yvs	UXAPPS LTD	5M+
	Voice Search Assistant	com.combo.voiceassistant	Standard Applications	1M+
	Voice Search: Search Assistant	com.prometheusinteractive.voice_launcher	Prometheus Interactive	10M+
	Voice Search	com.onlinehelp3011.VoiceSearch	V.K.D	100k+
	Voice Search	com.appybuilder.onlinehelp3011.BestVoiceSearch	Preeti Devi	1M+
	Voice Search	jp.gr.java_conf.mamama.voicesearchcw	AE App World	100K+
Other Type	**App Name**	**Package Name**	**Developer**	**Downloads**
Chatbot	ChatGPT	com.openai.chatgpt	OpenAI	50M+
	Google Gemini	com.google.android.apps.bard	Google	1M+
	Pi	ai.inflection.pi	Inflection AI	100K+
	ChatOn	ai.chat.gpt.bot	AIBY	5M+
	Luzia	co.thewordlab.luzia	Luzia	1M+
	Chatbot AI	com.codespaceapps.aichat	Codespace Dijital	5M+
Text-Voice Translate Assistant	AI Call Assistant & Screener	com.callassistant.android	Call Assistant Inc	50K+
	Speech Assistant AAC	nl.asoft.speechassistant	ASoft.nl	500K+
	Lookout-Assisted Vision	com.google.android.apps.accessibility.reveal	Google	500K+
	Speechify	com.cliffweitzman.speechify2	Speechify	1M+
Smart Home Assistant	Home Assistant	io.homeassistant.companion.android	Home Assistant	1M+
	Vision - Smart Voice Assistant	com.visionforhome	Turbo Trade S.A.	100K+
Spam Call Blocker	Truecaller: Spam Call Blocker	com.truecaller	Truecaller	1B+

Talks to play a talk about technology". Normally, task-executable voice assistants take voice input and translate it to text through Automatic Speech Recognition technique. This text is then processed with Natural Language Understanding (NLU) which is normally powered by pre-trained Large Language Models (LLMs), producing a series of operations to complete the specific task [7, 17]. After, they command the phone to perform operations (e.g., open the TED Talks app) and then convert the textual response back to speech using Text-to-Speech method.

These assistants go beyond simple query responses, integrating with multiple applications and performing multi-step operations based on user voice inputs. We first collect those voice assistants in the Google Play app store which is the most largest and the most accessible app market. The Google Play app store is often more transparent and friendly to academic research [33–35]. To ensure a comprehensive and controlled experimental environment, we utilized the searching feature of the Google Play app store to collect target applications. We created an initial set of search terms, including "*AI Assistant*", "*Voice Assistant*","*Smart Assistant*", "*AI Agent*", "*Smart Agent*","*Voice Agent*", *"Personal Assistant"*. To evaluate search results, we manually selected the top 20 results for each search query based on their default ranking. In addition to the direct results returned by the Google Play, we also collected *similar applications* that automatically recommended by the Google Play. After obtaining the initial set of results and removing duplicates, we narrowed down the task-executable voice assistant applications based on the following four criteria: 1)The application must support activation via voice commands as inputs; 2) The application must provide immediate verbal feedback upon receiving a command; 3) The application must execute meaningful operations beyond simple reading textual responses; 4) The application should have at least 50,000 downloads (*i.e.*, installs) on Google Play. This method allowed us to maintain a controlled and relevant set of applications for our experiments. Table 1 presents 10 identified task-executable voice assistant applications as the research targets in this paper.

In addition, as shown in the Table 1, voice assistant applications that do not meet the criteria can be empirically categorized into four groups based on their major functionality: "Chatbot", "Text-Voice Translate Assistant", "Smart Home Assistant", and "Spam Call Blocker". Among these categories, a significant portion of applications primarily function as chatbots. These applications are predominantly characterized by their ability to provide extended text-based responses and output by reading this content aloud. They cannot automate operations or perform complex tasks such as searching online by voice inputs. Other applications that are non task-executable can be categorized as different assistant tools, and each designed to deliver specific functionalities. Notably, some of these tools provide visual assistance by converting voice to text and text to voice, thereby enhancing accessibility for users. Additionally, certain applications specialize in blocking spam calls, thereby protecting users from unwanted communications. Above all, these voice assistant applications are not considered in this paper.

2.2 Characterization

Many studies have intensively investigated the voice assistants [4, 18, 21, 25, 46], whilst few have investigated the unique challenges brought by the task-executable feature. We first conduct an empirical investigation to assess their capabilities and understand

Table 2: The characterization of 10 voice assistants on seven different dimensions.

Name	Freeware	Registration	Battery Consumption	Data Consumption	RAM Occupation	Response Time	Recognize Accuracy	Robustness
Google Assistant	Free	Not Required	●	●	◐	●	◐	◐
Voice Access	Free	Not Required	●	N/A	●	○	●	●
Amazon Alexa	Free	Required	◐	◐	○	●	●	●
Ultimate Alexa Voice Assistant	Free&Paid	Required	◐	◐	○	○	●	●
Voice Search (UXAPPS)	Free	Not Required	◐	●	◐	○	●	●
Voice Search (V.K.D)	Free	Not Required	○	○	◐	○	●	●
Voice Search (Preeti Devi)	Free	Not Required	○	○	○	○	●	●
Voice Search (AE World)	Free	Not Required	◐	◐	◐	○	●	●
Voice Search Assistant	Free	Not Required	◐	○	○	◐	●	●
Voice Search: Search Assistant	Free&Paid	Not Required	◐	○	◐	○	●	●

their characteristics, facilitating our follow-up research toward their privacy and security risks. The comprehensive assessment is conducted on identified voice assistants along nine dimensions, and results are shown in the Table 2. The first two dimensions are their basic information. The next three dimensions target the efficiency of usage, including battery consumption, network data consumption, and phone ram occupation. The last three dimensions focus on the reliability of voice assistants, covering response time, voice to word recognize accuracy, and robustness. To ensure the consistency of evaluation, all evaluation is conducted on a Google's Pixel 7a, with 8GB memory and Android 14 operating system. We specifically define the eight dimensions as follows:

Freeware (1). This aspect denotes whether the voice assistant is available as a free service, a paid service, or a combination of both. We categorize the software into one of three groups: completely free, requiring payment for use, or offering both free and paid tiers. This evaluation helps determine the accessibility of the assistant for users with different budget considerations.

Registration (2). In this criterion, we evaluate whether the voice assistant requires user registration before use. We categorize the software as either requiring mandatory registration or allowing access without any registration process. This assessment is important for understanding the setup complexity and privacy implications for users.

Battery Consumption (3). Users of voice assistants frequently interact with their mobile phones, making battery consumption a primary concern for long-term usage. To simulate real-world usage, we selected ten input commands as follows, and interacted with the voice assistant once per minute for a total of ten minutes. Battery consumption was measured from the opening of the specific voice assistant application to the end of the last response. Voice assistants may utilize third-party applications (e.g., Google Search, speech recognition) to provide their functions. The battery consumption of these third-party applications is included in the total battery consumption of the voice assistant, as they are integral to the interaction. Consumption was monitored and collected by using AccuBattery[4], and the unit of measurement for this column is milliampere-hours (mAh). The voice assistants are scored as follows based on their battery consumption: The voice assistants are scored as (●) if their battery consumption is less than 20 mAh, if it is between 20 to 40, the coive assistant is marked as (◐); and if it is greater than 40, the voice assistant is marked as (○).

Ten Selected Input Commands

1.What is the weather in CITY.
2.What is the date today?
3.Open YouTube search for Baby Shark.
4.Give some music suggestion while doing sports.
5.What is the weather in the following week in CITY?
6.What is the trending movie?
7.Remind me on next Friday go to Supermarket at eight in the morning.
8.Set an alarm for tomorrow morning seven o'clock.
9.Suggest a restaurant in CITY for dinner.
10.Make a phone call to PERSON in the contacts.

Notes: CITY and PERSON are masked for the sake of anonymity.

Data Consumption (4). Voice assistants provide services using the Internet to access database and AI models, for purposes such as task planning, content searching, or accurate voice recognition. To monitor their mobile data usage during the ten commands test (as used for battery consumption), we employed GlassWire[5]. The data usage of third-party applications triggered by the voice assistant is also included in the total data consumption of the voice assistant. The data is calculated as the average usage across the ten tested

[4]https://accubatteryapp.com/
[5]https://www.glasswire.com/

commands, with the unit of measurement being kilobytes (KB). The voice assistants are scored based on their data consumption as follows. Voice assistants are scored as (●) if they only consumed data less than 500 KB, (◐) for 500 KB to 1,000 KB, and (○) for more than 1,000 KB.

RAM Occupation (5). The RAM occupation of applications on mobile phones plays an important role in stability. During the battery consumption test, we monitored the running services using the Android Developer tool and recorded the peak RAM occupation of the voice assistant. The RAM usage of third-party applications triggered by the voice assistant is also included, as these are integral to the voice assistant's functions. The measurement unit for RAM occupation is megabytes (MB). The voice assistants are scored based on their RAM usage as follows. We use (●) indicates the voice assistants occupies RAM below 200 MB and (◐) above 200MB to 400 MB; (○) for the voice assistants that 400 MB above.

Response Time (6). Response time of voice assistants is a critical factor of user experience. The processing time can vary depending on the complexity of the tasks. Here, we compute the average response time of the ten test cases as the final result. The measurement unit is seconds (s). The voice assistants are scored based on their average response time as follows. The voice assistant response time between 0 to 0.5 second marked as (●), if it is between 0.5 to 1 second, the voice assistant is marked as (◐); and if it is greater than 1 second, the voice assistant is marked as (○).

Recognize Accuracy (7). The speech recognition accuracy of voice assistants is essential to accurately execute the voice commands. When the user begins speaking and the voice assistant is activated, the recognized text are commonly displayed on the screen. To evaluate the accuracy, we conducted the assessment with the same ten predefined commands. For each command, we recorded whether the speech recognition accurately transcribed the spoken words into text. The scoring is based on the number of commands correctly recognized out of ten trials. If the correct times is ten in all ten times test with (●). More than five times (include five times) out of ten with (◐), and less than the five times out of ten with (○).

Robustness (8). The speech recognition accuracy of voice assistants across diverse accents serves as a crucial evaluation criterion. To assess this capability, AI-generated voice software, specifically Speechify[6] and LOVO[7], was employed to simulate three less prevalent English accents: Chinese English, Singaporean English, and Indian English. Five commands were randomly selected from a pool of ten predetermined input commands. These commands were then vocalized using the AI voice generators and subsequently played to the voice assistants under examination. The accuracy of speech recognition was determined by analyzing the text results displayed on the screen. A three-tier evaluation system was implemented to categorize the recognition accuracy: (●) denotes perfect recognition across all three accent variations without any issue, (◐) indicates partial recognition among the three accent variations could not be detected accurately, and (○) signifies complete failure to recognize all of the accent variations.

[6]https://speechify.com/
[7]https://lovo.ai/

As shown in Table 2, the comparison of 10 voice assistants across seven different dimensions reveals varying levels of performance. The majority of the voice assistants show high levels of performance in most dimensions. Google Assistant, Voice Access, and Amazon Alexa stand out, particularly in recognition accuracy and reliability, indicating their robustness. Meanwhile, some assistants like Voice Search (V.K.D) and Voice Search (Preeti Devi) exhibit lower performance, particularly in terms of response time and reliability. Regarding freeware status, most assistants are free, with a few like Ultimate Alexa Voice Assistant offering both free and paid versions. As for registration requirements, only Amazon Alexa and Ultimate Alexa Voice Assistant necessitate registration, while the others do not require it.

3 PRIVACY INCONSISTENCY BETWEEN SETTINGS, USAGE, AND DECLARATION (RQ1)

Privacy disclosures serve as the cornerstone of mobile privacy [23, 34, 35]. For the average user, privacy concerns are already significant and complex; however, for individuals who rely heavily on voice assistants, the issue becomes even more pronounced. Numerous studies have addressed the inconsistencies in privacy disclosures and actual implementation within mobile applications, including voice assistant (VA) apps [23, 33–35, 46]. However, these studies typically focus on bilateral inconsistencies and do not specifically examine task-executable VAs. In this section, we delve deeper into the inconsistencies between the following six dimensions, namely 1) Privacy Labels (i.e. Google Play Data Safety Section), 2) Permissions through actual usage, 3) Permissions identified by checker tools, 4) Permissions listed in mobile settings, 5) Disclosure in privacy policies, and 6) Declaration in menifest file. These different sources represent both developer-declared and user-observed permission practices. By systematically cross-referencing permissions from these sources, we conducted fifteen unique comparisons per application, allowing us to identify and quantify the status quo of inconsistencies across the different data sources and observations.

Google Data Safety: Google data safety are simplified and user-friendly summaries provided through Google Play [36], offering an overview of an application's privacy practices. Here the The Google Data Safety categories serve as a reference point for Android permissions and real-world usage patterns, facilitating the alignment of user-friendly descriptions with actual app behaviors.

Relevant Android Permission(s): Android permissions constitute a critical security mechanism within the Android operating system, regulating an application's access to sensitive device resources and user data [15]. The permissions are matched with permission groups [29, 38] as illustrated in Table 3, facilitating a semantic alignment between permission groups and data safety categories. In instances where data safety types lack a direct correspondence with related Android permissions, this is denoted by "-" in the table. For example, the "Name" category in data safety, defined as "How a user refers to themselves, such as their first or last name, or nickname," does not have an associated Android permission for users referring to themselves, resulting in no linked permission. Also for the permission with superset, the original permission will be picked only. For example, the BLUETOOTH permission allows an app to

Table 3: The mapping between Google Data Safety types, Android permissions, and testing commands.

Google Data Safety Type	Relevant Android Permission(s)	Testing Command
Approximate location	ACCESS_COARSE_LOCATION	-
Precise location	ACCESS_FINE_LOCATION	Show my location
Phone number	READ_PHONE_STATE	
Fitness info	ACTIVITY_RECOGNITION	-
SMS or MMS	READ_SMS	Send message to "myself", what is going on.
Photos	READ_EXTERNAL_STORAGE	-
	CAMERA	
Videos	READ_EXTERNAL_STORAGE	-
	CAMERA	
Voice or sound recordings	RECORD_AUDIO	-
	READ_VOICEMAIL	Read voice mail.
Music files	READ_EXTERNAL_STORAGE	-
Other audio files	READ_EXTERNAL_STORAGE	-
Files and docs	READ_EXTERNAL_STORAGE	-
	MANAGE_EXTERNAL_STORAGE	Manage external storage.
Calendar events	READ_CALENDAR	-
	WRITE_CALENDAR	Add meeting at 2 o'clock on tomorrow.
Contacts	READ_CONTACTS	Call "myself" in contacts.
	WRITE_CONTACTS	Add contacts with phone number 222 an name is Dam.
	GET_ACCOUNTS	Show accounts number.
Installed apps	QUERY_ALL_PACKAGES	-
Diagnostics	BATTERY_STATS	Check battery stats.
Other app performance	CLEAR_APP_CACHE	Clear Youtube cache.
Device or other IDs	READ_PHONE_STAT	-
	BLUETOOTH	Pair bluetooth with headphone.

perform basic Bluetooth communication, such as connecting to paired Bluetooth devices. The BLUETOOTH_ADMIN permission, on the other hand, grants more extensive control over Bluetooth functionality, including the ability to discover and pair with new devices, as well as modify Bluetooth settings. BLUETOOTH_ADMIN is considered a superset of BLUETOOTH, meaning that an app with BLUETOOTH_ADMIN permission implicitly has all the capabilities granted by the BLUETOOTH permission, plus additional administrative functions.

Testing Command: Voice commands are utilized to detect the permission usage of task-executable voice assistants. The instructions required for specific permissions are approached as semantically as possible. For example, to testing the READ_VOICEMAIL permission, here by using "Read voice mail". This methodology aims to establish a correlation between voice assistant's actual usage of related permission with corresponding google data safety.

Privacy disclosures are crucial in informing users about how their data is collected, processed, and shared, fostering transparency and trust. The research focuses on voice assistants capable of executing actions upon receiving commands and associating security and privacy issues. Consequently, only permissions and data types that provide users with specific functional awareness are selected for analysis. For example, the "Precise location" would count into the consideration, but "Purchase history" will not be considered. A mapping relation between Google Data Safety types, Android permissions, and testing commands are listed in the Table 3. To understand the extent of privacy declaration inconsistency, we analyzed the six sources mentioned and cross-compared them to uncover any discrepancies. And we define these six dimensions specifically as follows:

Permissions Listed in Google Play's Data Safety Section (1). This section focuses on analyzing the permissions that an application lists in the Data Safety [36] section of its Google Play Store listing. The Data Safety section is a relatively feature introduced by Google to provide users with transparent information about how their data is handled by apps available on the Play Store. This intended to help users make informed decisions about app downloads by presenting the types of data collected. For the purposes of our research, we are treating the data safety declaration as authoritative and accurate.

Permission Identified Through Actual Usage (2). This data source involves a controlled testing (Table **??**) approach to determine which permissions an Android application actually utilizes during operation. The methodology employs a predefined set of commands or actions designed to trigger various app functionalities that are likely to require specific permissions. We use Table 3 to match permissions with their corresponding data safety declarations.

Permission Identified Through a Permission Checker Tool (3). Here we use the Permission Pilot [43], a specialized application designed to analyze and audit the permissions requested by Android applications. Permission Pilot offers a detailed inspection of an app's permission requests, providing insights into what each app is capable of accessing on a user's device. The purpose of comparing the permissions detected by the Permission Pilot application with other data sources is to assess the reliability of the permission checker tool. By evaluating how well the permissions identified by Permission Pilot align with those observed through other means,

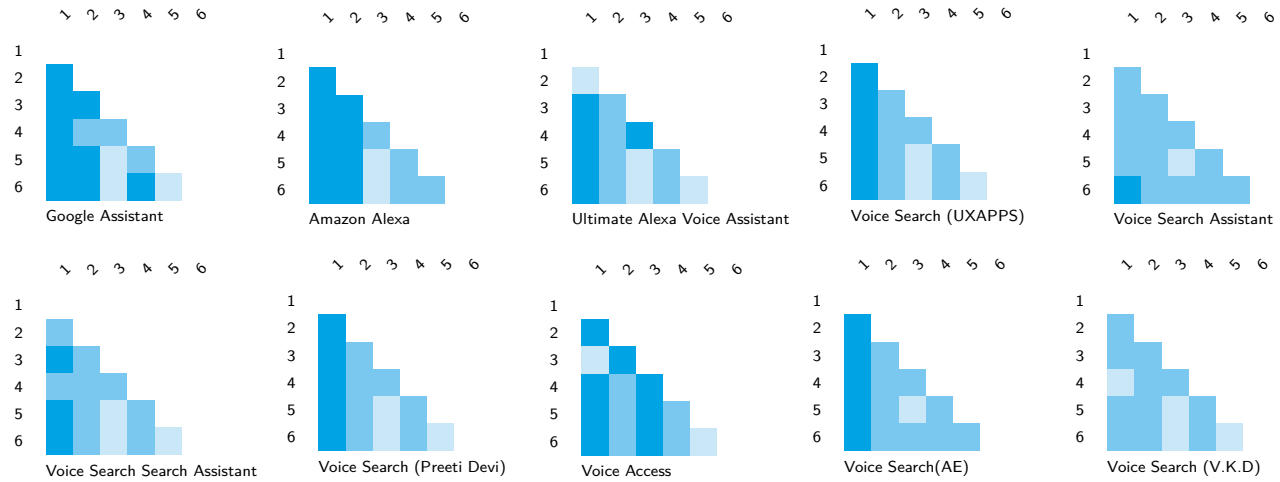

Figure 1: The level of inconsistency for each dimension for 10 VAs. A deeper color indicates a higher degree of inconsistency in the privacy disclosures. The number denotes the count of inconsistent permissions/data types. (color 0-2 , color 2-5, color 5+)

such as direct examination of the Android manifest or the permissions listed in device settings. We use Table 3 to match permissions with their corresponding data safety declarations.

Permission List Observed in Android Settings (4). Each application in Android is using permission list for the permission that it required for running. It is a comprehensive overview of the permissions that an Android application has requested and been granted on a user's device. This list is visible within the device's settings, typically under the "Apps" section, where users can see which permissions an app has access to, such as location, camera, contacts, storage, and more. Each permission represents a specific capability that the app can use to interact with the device's hardware or data, ranging from accessing the microphone to reading the user's calendar events. We firstly match the setting with related android permission then use Table 3 to match permissions with their corresponding data safety declarations.

Permissions Declared in the Application Privacy Statement (5). For this analysis, we employ the PoliGraph tool [10]. This tool is specifically designed to extract and compile information about permissions mentioned in software applications' privacy statements. We then use Table 3 to match these permissions with their corresponding data safety declarations. This approach allows us to verify that an app's privacy statement accurately and transparently discloses the permissions it uses, which is crucial for maintaining user trust and ensuring regulatory compliance.

Permission from Manifest (6).This data source involves directly examining an Android application's manifest file (AndroidManifest.xml) to identify all permissions declared by the app developer. The Android manifest is a crucial XML file that provides essential information about the app to the Android system, including the permissions the app requires to function[8]. We use Table 3 to match these identified permissions with their corresponding data safety declarations.

Figure 1 shows the examination results. Under reporting of Permissions: Google Play's Data Safety section often omits permissions that are detected through static analysis (by permission checker tools) or observed in the manifest file. For example, certain permissions related to sensitive data, such as location or microphone access, were found in the manifest file but not listed in the Data Safety section. Additionally, there were cases where the privacy statement of the app claimed compliance with privacy standards, yet the actual permissions requested and used by the app contradicted those claims. For instance, permissions that were declared as not being used or necessary in the privacy statement were, in fact, actively requested and used by the app, leading to an over-disclosure issue. These discrepancies pose significant risks to user privacy, as they highlight a lack of alignment between what is declared and what is actually practiced. The failure to adequately disclose permissions, especially in the Google Play Data Safety section, undermines user trust and may result in uninformed consent, where users are unaware of the full extent of data collection and sharing. Furthermore, the presence of over-declared or under-declared permissions exposes apps to potential regulatory scrutiny, as this misalignment can be interpreted as non-compliance with privacy regulations like GDPR or CCPA.

> **Finding 1:** The discrepancies between privacy disclosure, settings, and actual usage pose significant risks to user privacy, especially those heavily rely on VAs. The failure to adequately disclose permissions, especially in the Google Play Data Safety section, undermines user trust and may result in uninformed consent, where users are unaware of the full extent of data collection and sharing.

[8]https://developer.android.com/guide/topics/manifest/manifest-intro

4 RQ2: PRIVACY MISDISCLOSURE IN MEGA APPS

As we delved deeper into the ecosystem of voice assistants, we discovered that many voice-based functionalities do not exist as standalone apps. Instead, they are packaged as mini apps that are integrated and invoked within larger, more comprehensive apps (*i.e.,* Mega Apps). For example, in the case of Alexa, it provides skill kits (*a.k.a.,* skills) to deliver voicebased actions, including collecting health data, making phone calls to contacts, etc. Unfortunately, we found that despite these Mega Apps (e.g., Alexa) being required by regulations to declare all permissions associated with their integrated functionalities in their data safety sections, they often fail to do so. In other words, Mega Apps frequently overlook the declaration of data safety for these linked functionalities, which is a violation of regulatory requirements. This oversight may not only misleads users but also leaves them uninformed about potential privacy risks, making it difficult to trace the source of data exposure in the event of a privacy breach.

As a showcase presented in Figure 2, we present in the following a concrete example to demonstrate its implications for user privacy and the potential risks associated with undeclared permissions in Mega Apps. The diagram illustrates a scenario within the Amazon Alexa app, where voice inputs trigger the invocation of various "Alexa skills", allowing users to access sensitive private information, such as health data, fitness info, etc. Specifically, when a voice command *"Alexa, ask Fitbit how many steps I've taken"* is issued, the fitness data is accessed and returned to the user with a response *You've taken 6,000 steps today.*. However, the critical issue identified is that no explicit permission for accessing fitness data has been granted to the Alexa application and what's even worse the Alexa didn't claim privacy acquisition in the online app market such as Google Play [28].

By looking into this case, we find out that there is a alarming issue where privacy declarations within skills are frequently overlooked and not adequately reflected in the corresponding mega app, leading to violations of data safety regulations, such as GDPR and CCPA. In other words, these sensitive behaviors within the skills are not disclosed in the mega app's privacy and data safety declarations, resulting in users being unaware that the mega app might invoke these behaviors when using voice commands. This lack of transparency raises significant privacy concerns and potential regulatory violations.

To address these privacy concerns, voice assistant platforms (e.g., Amazon Alexa) should adopt more transparent and comprehensive data safety practices, covering not only the native code (i.e., code written by the app's developers) but also integrated skills and third-party libraries. There is an urgent need to explicitly and thoroughly declare any privacy collection and processing within the mega app, ensuring users are fully aware of how their sensitive information is handled. Regulatory frameworks should also enforce stricter requirements for data safety declarations, mandating that platforms clearly specify all relevant data categories. By improving transparency and aligning privacy declarations with actual data practices, users can be better informed about who accesses their data and for what purposes, ultimately enhancing trust and compliance in the digital health ecosystem.

Finding 2: The privacy policies of mega apps often overlook the disclosure of privacy usage related to their mini apps (e.g., skills) or integrated functionalities. This lack of proper declaration can result in ineffective privacy policies, undermining user trust and potentially leading to regulatory violations.

5 RQ3: *THE FAIL OF MAGINOT LINE*: PRIVILEGE ESCALATION THROUGH INTER-APP INTERACTIONS

Security issues are a key focus of voice assistant app analysis, including voice squatting, voice app faking [13], voice masquerading [48], etc. These identified security issues often come with bad code practices in voice app development, leading to a widespread vulnerability among voice apps under attack [16, 21, 39].

Despite the emergence of various attack mechanisms, privilege escalation in voice apps has not been studied before. In this work, we observed a privilege escalation mechanism through inter-app interactions, as illustrated in Figure 3. In this scenario, the host app does not declare any dangerous permissions (e.g., "android.permission.READ_PHONE_STATE", "android.permission.CALL_PHONE", "android.permission.READ_CONTACTS") in its Manifest file, yet it can still trigger sensitive behaviors by delegating them to the client app through inter-app interactions. This attack mechanism is considered as privilege escalation because the host app fails to declare the appropriate permissions, allowing dangerous behaviors to be triggered without the user's consent. This inter-app interaction mechanism can be exploited by attackers to bypass permission declarations by invoking dangerous behaviors from other apps, thereby leading to a privilege escalation attack.

Specifically, when a voice command is given to the host app[9], such as "Alexa, ask Phone Link to call A[10]", the host app wakes up the client app[11] to execute the action, where the corresponding permissions are granted. Here, we perform reverse engineering on the host and client APK, inspecting the code with a focus on its logic and security implications, as shown in listing 1 and listing 2. From the host app's perspective, there is a *BackgroundListen*(line 4 in listing 1) class, within which the *AvsService* is invoked (line 10 in listing 1) to process audio command and then send corresponding message to cloud service (*Firebase Cloud Messaging*[12]) for further actions. This method also requests the "android.permission.RECORD_AUDIO" permission (line 11 in listing 1) using because it requires access to the device's microphone to capture audio. Then, on the side of the client app (listing 2), a custom-designed class *MessagingService* which extends *FirebaseMessagingService* to handle specific intents from Firebase messages and perform corresponding actions based on those intents. Specifically, the *dispatchMessage* method processes

[9] https://play.google.com/store/apps/details?id=com.customsolutions.android.alexa&hl=en_AU

[10] A can be any name in the contact list

[11] https://play.google.com/store/apps/details?id=com.customsolutions.android.phonelink&hl=en_AU

[12] Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that enables developers to send messages to users across different devices and platforms, including Android, iOS, and web applications : https://firebase.google.com/docs/cloud-messaging

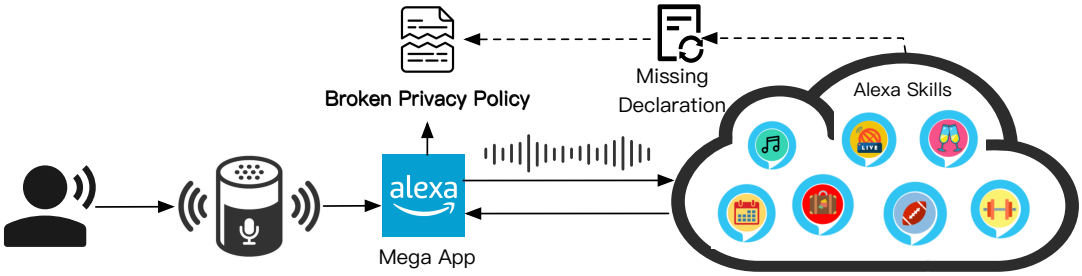

Figure 2: Violate Data Safety Declaration of Mega Application

the incoming remote message and extracts the intent information from it. If the intent is "PlaceCallInten", it calls the *makePhoneCall* method (lines 8 in listing 2) to make the phone call. Inside the *makePhoneCall* method, it verifies whether the app has been granted the necessary permissions *READ_PHONE_STATE, CALL_PHONE, READ_CONTACTS* (lines 14 to 17 in listing 2) and also checks if the app can draw overlays on the screen (line 19 in listing 2), which is required for certain versions of Android (below API 29). Then, it retrieves contact information (lines 22 to 24 in listing 2) and proceeds to make the phone call (lines 27 to 31 in listing 2).

Such privilege escalation, particularly when involving the misuse of sensitive permissions like *android.permission.CALL_PHONE*, poses a significant threat to both user security and privacy. This form of attack allows malicious attackers to elevate their access rights, potentially enabling unauthorized actions such as making calls, accessing personal data, or manipulating system settings without the user's consent. The implications can be quite severe: not only can attackers exploit this mechanism to conduct fraudulent activities, but they can also undermine the integrity of the entire mobile ecosystem, leading to a breach of trust between users and service providers. In essence, improper management of such permissions is a gateway to broader systemic risks, highlighting the urgent need for rigorous permission handling and vigilant security practices within application development.

```
1  // Simplified Code Snippet of Host App
2  package com.customsolutions.android.alexa;
3
4  public class BackgroundListen extends AlexaActivity{
5      ...
6  @Override
7  public void onStart() {
8     ...
9     //AvsService is responsible for conducting natural language
          data processing and sending messages to Firebase Cloud
          Messaging (i.e., Google Cloud Messaging).
10    AvsService.sendMessage(...);
11    ActivityCompat.requestPermission(this,
        "android.permission.RECORD_AUDIO", ...);
12 }}
```

Listing 1: Code example demonstrating the behavior of host app. The code snippet is extracted from app *Ultimate Alexa Voice Assistant*.

```
1  // Simplified Code Snippet of Client App
2  package com.customsolutions.android.phonelink;
3  public class MessagingService extends FirebaseMessagingService
       {
4      ...
5   public final void dispatchMessage(RemoteMessage
        remoteMessage) {
6   //string5 is the intent info extracted from Message
7   if (string5.equals("PlaceCallIntent")) {
8        makePhoneCall(...);
9        return;
10 }}
11
12 public final void makePhoneCall(...){
13  ...
14  if(checkSelfPermission(
15  "android.permission.READ_PHONE_STATE",
16  "android.permission.CALL_PHONE",
17  "android.permission.READ_CONTACTS")
18  &&
19  (Build.VERSION.SDK_INT < 29 ||
        Settings.canDrawOverlays(...))){
20      ...
21  //Retrieve Contacts
22     Cursor query = contentResolver.query(Contacts.CONTENT_URI,
         ...);
23     Person p = new Person();
24     p.phoneNumber = query.getString(...);
25     ...
26  //Make Phone call
27     StringBuilder sb = new StringBuilder();
28     sb.append("tel:").append(p.phoneNumber);
29     Intent intent = new Intent("android.intent.action.CALL",
30     Uri.parse(sb.toString()));
31     startActivity(intent);
32 };}}
```

Listing 2: Code example demonstrating the behavior of client app. The code snippet is extracted from app *Phone Link: Skill For Alexa*.

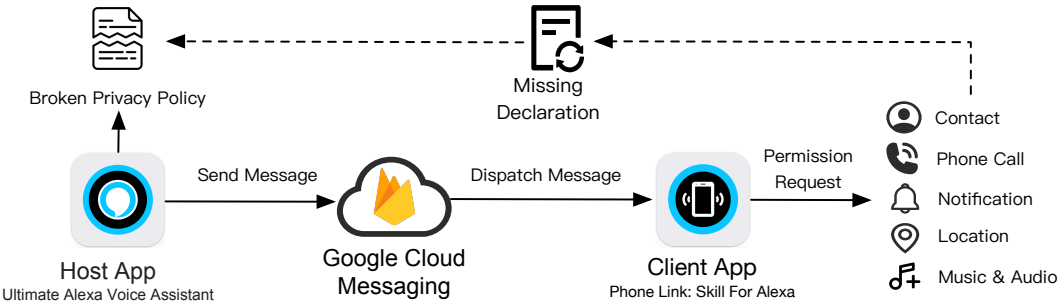

Figure 3: The Working Mechanism of Privilege Escalation Through Inter-app Interactions

Finding 3: Our finding reveals a sophisticated privilege escalation attack model that exploits inter-app interaction mechanisms. This attack model leverages the inherent pathways between applications to escalate privileges without user consent. The discovery underscores that inter-app communications can be weaponized to compromise the integrity and security of the device, allowing attackers to gain unauthorized access to sensitive data.

6 RQ4: THE ABUSE OF GOOGLE SYSTEM APPLICATION

To date, Google Assistant serves as a key platform enabling voice-forward control of Android apps. It allows developers to implement custom functionalities through voice commands using Google Assistant's framework. Fewer security issues have been identified in Google Assistant compared to Amazon Alexa, largely due to its limited transparency as a closed-source application. This hinders the discovery of potential security vulnerabilities, leaving many threats unexposed.

In this section, we introduce a novel mechanism through which developers can exploit Google System Services to achieve privilege escalation. As illustrated in Figure 4, this process involves an application, referred to as "Voice Search"[13], interacting with a Google system application named as "com.google.android.tts"[14]. Specifically, when an audio command is input to "Voice Search", it is transmitted to the Google system application for natural language processing, converting the audio command into text. This text command is then executed directly within the Google system application, which allows it to access and retrieve sensitive information such as email addresses, device IDs, diagnostic logs, and other private data. Notably, this data access occurs without explicit permission declarations from the "Voice Search" application, thereby contravening data protection regulations mandated by international standards, such as the General Data Protection Regulation (GDPR) [12]. In summary, we have identified a new mechanism enabling developers to invoke sensitive behaviors (i.e., those typically governed by permissions) by leveraging Google System Services, rather than declaring standard permission controls within the host application. The absence of explicit permission requests from the host application compounds the issue by bypassing user consent mechanisms designed to protect their information. This not only undermines user trust in the applications but also potentially leads to legal and regulatory repercussions for developers and service providers who fail to adhere to data protection laws.

```
1  // Simplified Code Snippet of Voice Search
2  package com.onlinehelp3011.VoiceSearch;
3  public class MainActivity extends AppCompatActivity {
4    ...
5    // Invoke google system application for speech recognize.
6    public void btnvoice(View view) {
7     Intent intent = new
          Intent("android.speech.action.RECOGNIZE_SPEECH");
8     startActivityForResult(intent, 100);
9     ...
10 }}
```

Listing 3: Code example demonstrating the behavior of Voice Search. The code snippet is extracted from application *com.onlinehelp3011.VoiceSearch*.

```
1  // Simplified Code Snippet of Speech Recognition & Synthesis
       (google system application)
2  package defpackage;
3  public final class fbq {
4   public static ComponentName a(Context context) {
5    ResolveInfo resolveInfo =
         packageManager().queryIntentActivities( new
         Intent("android.speech.action.RECOGNIZE_SPEECH");
6    if (packageName.equals("com.google.android.tts") &&
7        Build.VERSION.SDK_INT >= 31) {
8        return new ComponentName(packageName, activityName);
9  }}}}
```

Listing 4: Code example demonstrating the behavior of Speech Recognition & Synthesis. The code snippet is extracted from application *com.google.android.tts*.

When the user click the voice input button on application[15], the host app wakes up the google system application[16] to recognize the speech, where text result will return and do the google search. Here, we perform reverse engineering on the host application APK and client application APK, inspecting the code with a focus on its logic and security implications, as shown in listing 3 and listing 4. From the host application's perspective, there is a *btnvoice* (line 6 in listing 3) class, within which the *android.speed.action.RECOGNIZE_SPEECH* is invoked (line 5 in listing 4) to make google system application proceeding the voice input. On the side of the client application (listing 4), a custom-designed class *defpackage* to handle

[13]https://play.google.com/store/apps/details?id=com.onlinehelp3011.VoiceSearch
[14]https://play.google.com/store/apps/details?id=com.google.android.tts
[15]https://play.google.com/store/apps/details?id=com.onlinehelp3011.VoiceSearch&hl=en_AU
[16]https://play.google.com/store/apps/details?id=com.google.android.tts&hl=en_AU

specific intents from host application and record voice command which requiring the "android.permission.RECORD_AUDIO" as it needs access to the device's microphone to capture audio.

In addition, detecting such data flows presents a significant challenge for program analysis, as the Google system applications involved are often closed-source and highly obfuscated. The closed-source nature and obfuscation severely impairs the ability of security analysts and automated tools to scrutinize the internal workings of these system applications.

This mechanism is critical to end users because the lack of privacy transparency in the voice assistant app often results in unauthorized actions without appropriate permission declarations. Additionally, we argue that this mechanism can be exploited by attackers for malicious purposes, leaving users vulnerable to privacy breaches and data misuse.

To mitigate these privacy risks, voice assistant applications should implement more transparent data protection practices. Specifically, they need to explicitly disclose any involvement of system-provided services and clearly outline their data handling procedures. Regulatory frameworks should also impose stricter disclosure requirements to ensure users are fully informed about all entities accessing their data and the precise permissions granted. By standardizing privacy declarations across all integrated services, users can regain trust in how their data is managed, leading to improved privacy outcomes in the interconnected landscape of voice assistants.

Finding 4: Our findings reveal a critical privacy vulnerability in voice assistant applications stemming from their interaction with system-provided services. This issue arises from the lack of transparency and explicit permission declarations, which can be exploited to perform unauthorized actions without user consent. The discovery highlights the need for improved privacy practices and regulatory measures to ensure users are fully informed and protected against potential data misuse and security breaches.

7 DISCUSSIONS AND IMPLICATIONS

Based on the analysis of the 10 selected voice assistants, a summary of key findings related to the research questions has been compiled (Table 4). For Research Question 1, addressing disclosure and inconsistency among settings, usage, and declarations, a three-tier classification system is employed in the heat map (Figure 1). Instances with more than five significant discrepancies are denoted by (●), those with two to five discrepancies by (◐), and those with fewer than two by (○). Research Question 2 examines the mis-declaration of permissions in mega applications. (●) indicates applications exhibiting this issue and (○) represents those without. For Research Question 3, which investigates cross-application interactions with undeclared permission usage, (●) denotes applications manifesting this problem and (○) signifies those that do not. Research Question 4 focuses on innovations within Google system applications. (●) signifies the presence of an innovation and (○) indicates its absence.

Challenges and opportunities co-exist in the current voice assistant applications. Based on our observations and study results, we summarize some findings for various roles or stockholders in the voice assistant ecosystem.

Usability considerations The usability of voice assistant applications must be enhanced to empower users in managing their privacy effectively while maintaining the convenience and accessibility these technologies offer. Our study highlights a significant challenge: the complexity of privacy management in voice assistants often overwhelms users, leading to potential privacy risks. To address this, voice assistant developers and platform providers should prioritize the design of intuitive, user-friendly privacy controls that clearly communicate data practices and allow for granular permission management. This approach should include simplified yet comprehensive privacy notifications, easy-to-understand data flow visualizations, and streamlined processes for reviewing and modifying privacy settings. Moreover, developers should strive to balance functionality with privacy, offering users clear choices about data sharing without compromising core features.

App developer While app developers are developing new voice assistant applications, they should be aware of the data safety declaration in the Google Play Store. As illustrated in research question 2, the data safety declared on google play have different with actual privacy usage. Some do not directly declare specific data usage, such as health data and reading contacts, and some through other applications,i.e., the invoke of google original application, and third-party applications. Since voice assistant developers must make sure the data safety they provide on the google play must be comprehensive, and thoughtful, because this is the first place where users come into contact with other applications.

Voice assistant platform providers Our analysis suggests voice assistant (the mega apps) providers should work on improving recognised data safety use, since the majority of mega voice assistants on the markets only provide a very limited scope of data safety. In research question 3, the first two research cases pointing out two separate questions for mega voice assistant platform: **a)** Miss-declaration of it's own "skill" data safety, and **b)** playing a role as a third party to help the application jump to other application that take more permission than previous application.

Privacy regulators Privacy regulators must address the unique challenges posed by task-executable voice assistants to ensure user privacy and data protection in this rapidly evolving ecosystem. Our findings reveal critical gaps in current voice assistant security practices and privacy declarations, necessitating a proactive regulatory approach. Regulators should focus on developing comprehensive guidelines that address the complex data flows in voice assistant systems, including integrated skills and third-party functionalities. These guidelines should mandate transparent and accurate declaration of all permissions and data access, including those invoked through system applications or inter-app interactions. Additionally, regulators should establish clear standards for privacy declarations in mega apps, addressing the potential for privilege escalation and ensuring that users are fully informed about how their data is collected, used, and shared across the voice assistant ecosystem.

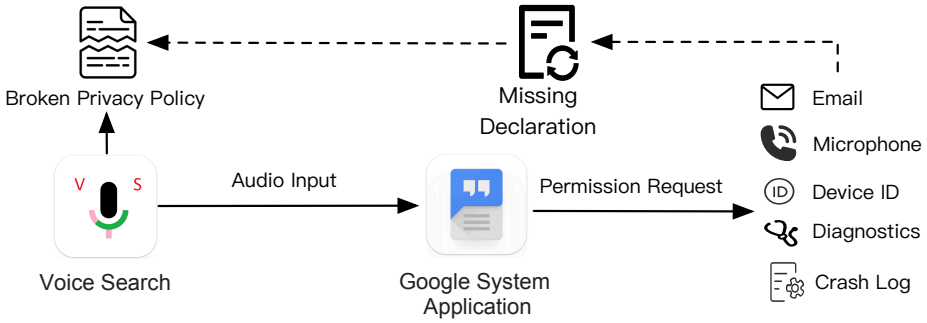

Figure 4: The Working Mechanism of Privilege Escalation Through Google System Application

Table 4: The security and privacy performance of 10 examined voice assistants on research questions. Black dots denote the violation exists.

Name	RQ1	RQ2	RQ3	RQ4
Google Assistant	●	●	○	○
Voice Access	◐	○	○	○
Amazon Alexa	●	●	○	○
Ultimate Alexa Voice Assistant	◐	○	●	○
Voice Search (UXAPPS)	◐	○	○	●
Voice Search (V.K.D)	○	○	○	●
Voice Search (Preeti Devi)	◐	○	○	●
Voice Search (AE World)	◐	○	○	●
Voice Search Assistant	○	○	○	●
Voice Search: Search Assistant	◐	○	○	●

8 RELATED WORK

8.1 Voice Assistants Privacy Compliance

Xie et al. [46] developed Skipper, a tool that automatically analyzes privacy policies of Amazon Alexa skills using NLP and machine learning techniques to detect non-compliance issues. Yan et al. [47] proposed QuPer, which assesses privacy policies based on four metrics: timeliness, availability, completeness, and readability. Liu et al. [22] introduced Elevate, a model-enhanced LLM-driven VUI testing framework for VPA apps. It combines LLMs with behavioral models for improved testing coverage. Liao et al. [24, 25] conducted comprehensive studies on privacy compliance in VPA apps, including GDPR compliance in European marketplaces and large-scale analysis of privacy policies in US marketplaces. They developed SkillScanner, a static code analysis tool to detect privacy and policy violations in skill development, and SkillPoV, a tool for generating voice-based privacy notices.

8.2 VAs Security Risks

Alrawi et al. [4] conducted a comprehensive survey of IoT security, proposing a four-pronged classification model for analyzing IoT security. Sivaraman et al. [41] explored network-level security for smart home IoT devices, highlighting the importance of securing the communication between devices. Zhang et al. [49] investigated privacy issues in smart homes, focusing on information leakage through encrypted IoT traffic. In the context of voice-controlled IoT, Ding et al. [11] discovered new vulnerabilities in the Amazon Alexa platform, demonstrating how malicious third-party developers could hijack built-in voice commands to invoke malicious IoT skills. Their work revealed potential security risks in IoT skills that had been previously underexplored.

8.3 Android Program Analysis

Liu et al. [27] conducted the first preliminary study on security risks of Android TV apps, revealing various security issues similar to those found in smartphone apps. Arzt et al. [5] developed FlowDroid, a precise context, flow, field, and object-sensitive taint analysis tool for Android apps. These techniques have been applied to detect various security issues, including over-privilege [15], unauthorized data access [50], insecure data transmission [14], and component hijacking [8].

While existing studies provide significant insights into privacy and security risks, they often overlook the unique challenges posed by task-executable voice assistants. Our research addresses this gap by offering a timely and comprehensive investigation into these challenges.

9 CONCLUSION

This paper presents the first comprehensive empirical study on privacy and security risks in Android task-executable voice assistants, analyzing 10 mainstream applications and uncovering significant vulnerabilities. Our market penetration analysis reveals that 20 percent of top voice assistant apps on Google Play are task-executable, with Google Assistant leading at over 1 billion downloads. We identified substantial noncompliance with privacy laws and frequent under-declaration of permissions, particularly in mega apps integrating third-party functionalities. Our privacy declaration analysis exposed significant discrepancies across various data sources, highlighting a critical lack of transparency in data handling practices. Notably, we uncovered three novel security vulnerabilities: privacy misdeclaration in mega apps, a privilege escalation attack model exploiting inter-app interactions, and an abuse mechanism involving Google System Applications. These findings collectively underscore the urgent need for improved privacy practices and enhanced security measures in voice-based interfaces. Our permissions coverage analysis revealed that 60 percent of analyzed apps under-declare

permissions, averaging 3.5 undeclared permissions per app. In summary, both users and developers face substantial risks of privacy breaches and potential regulatory violations when employing voice assistant applications without careful consideration.

9.1 Future Work

The future work could prioritize on the development of comprehensive automated testing frameworks specifically designed for task-executable VAs. These frameworks are crucial to address the complex vulnerabilities and privacy issues uncovered in our study. Such automated testing should encompass dynamic interaction simulation to cover a wide range of voice commands and user scenarios, continuous privacy compliance checking to ensure adherence to evolving regulations, and inter-app communication analysis to prevent privilege escalation attacks.

10 DATA AVAILABILITY

To support reproducibility in the research community, our replication package is available at: https://anonymous.4open.science/r/TaskExecutableVoiceAssistant-27AE/

REFERENCES

[1] Luca Hern'andez Acosta and Delphine Reinhardt. A survey on privacy issues and solutions for voice-controlled digital assistants. *Pervasive and Mobile Computing*, 80:101523, 2022.

[2] Efthimios Alepis and Constantinos Patsakis. Monkey says, monkey does: security and privacy on voice assistants. *IEEE Access*, 5:17841–17851, 2017.

[3] Nourah AlOtaibi and Federico Lombardi. Privacy and security evaluation of amazon echo voice assistant. In *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*, pages 1–6. IEEE, 2021.

[4] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. Sok: Security evaluation of home-based iot deployments. In *2019 IEEE symposium on security and privacy (sp)*, pages 1362–1380. IEEE, 2019.

[5] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. *ACM sigplan notices*, 49(6):259–269, 2014.

[6] Tom Bolton, Tooska Dargahi, Sana Belguith, Mabrook S Al-Rakhami, and Ali Hassan Sodhro. On the security and privacy challenges of virtual assistants. *Sensors*, 21(7):2312, 2021.

[7] Peng Cheng and Utz Roedig. Personal voice assistant security and privacy—a survey. *Proceedings of the IEEE*, 110(4):476–507, 2022.

[8] Erika Chin, Adrienne Porter Felt, Kate Greenwood, and David Wagner. Analyzing inter-application communication in android. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 239–252, 2011.

[9] Leigh Clark, Philip Doyle, Diego Garaialde, Emer Gilmartin, Stephan Schlögl, Jens Edlund, Matthew Aylett, João Cabral, Cosmin Munteanu, Justin Edwards, et al. The state of speech in hci: Trends, themes and challenges. *Interacting with computers*, 31(4):349–371, 2019.

[10] Hao Cui, Rahmadi Trimananda, Athina Markopoulou, and Scott Jordan. {PoliGraph}: Automated privacy policy analysis using knowledge graphs. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 1037–1054, 2023.

[11] Wenbo Ding, Song Liao, Long Cheng, Xianghang Mi, Ziming Zhao, and Hongxin Hu. Command hijacking on voice-controlled iot in amazon alexa platform. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 654–666, 2024.

[12] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council.

[13] Luise Frerichs Fabian Bräunlein. Smart spies: Alexa and google home expose users to vishing and eavesdropping.

[14] Sascha Fahl, Marian Harbach, Thomas Muders, Lars Baumgärtner, Bernd Freisleben, and Matthew Smith. Why eve and mallory love android: An analysis of android ssl (in) security. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 50–61, 2012.

[15] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. Android permissions demystified. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 627–638, 2011.

[16] Nathaniel Fruchter and Ilaria Liccardi. Consumer attitudes towards privacy and security in home assistants. In *Extended Abstracts of the 2018 CHI conference on human factors in computing systems*, pages 1–6, 2018.

[17] Matthew B Hoy. Alexa, siri, cortana, and more: an introduction to voice assistants. *Medical reference services quarterly*, 37(1):81–88, 2018.

[18] Yue Huang, Borke Obada-Obieh, and Konstantin Beznosov. Amazon vs. my brother: How users of shared smart speakers perceive and cope with privacy risks. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–13, 2020.

[19] Trung Dong Huynh, William Seymour, Luc Moreau, and Jose Such. Why are conversational assistants still black boxes? the case for transparency. In *Proceedings of the 5th International Conference on Conversational User Interfaces*, pages 1–5, 2023.

[20] Veton Kepuska and Gamal Bohouta. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pages 99–103. IEEE, 2018.

[21] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. Alexa, are you listening? privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. *Proceedings of the ACM on human-computer interaction*, 2(CSCW):1–31, 2018.

[22] Suwan Li, Lei Bu, Guangdong Bai, Fuman Xie, Kai Chen, and Chang Yue. Model-enhanced llm-driven vui testing of vpa apps. *arXiv preprint arXiv:2407.02791*, 2024.

[23] Tianshi Li, Kayla Reiman, Yuvraj Agarwal, Lorrie Faith Cranor, and Jason I Hong. Understanding challenges for developers to create accurate privacy nutrition labels. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*.

[24] Song Liao. Ensuring the privacy compliance of voice personal assistant applications. 2024.

[25] Song Liao, Mohammed Aldeen, Jingwen Yan, Long Cheng, Xiapu Luo, Haipeng Cai, and Hongxin Hu. Understanding gdpr non-compliance in privacy policies of alexa skills in european marketplaces. In *Proceedings of the ACM on Web Conference 2024*, pages 1081–1091, 2024.

[26] Song Liao, Christin Wilson, Long Cheng, Hongxin Hu, and Huixing Deng. Measuring the effectiveness of privacy policies for voice assistant applications. In *Proceedings of the 36th Annual Computer Security Applications Conference*, pages 856–869, 2020.

[27] Yonghui Liu, Li Li, Pingfan Kong, Xiaoyu Sun, and Tegawendé F Bissyandé. A first look at security risks of android tv apps. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*, pages 59–64. IEEE, 2021.

[28] Amazon Mobile LLC. Amazon alexa. https://play.google.com/store/apps/datasafety?id=com.amazon.dee.app&hl=en_AU.

[29] Ryan McConkey and Oluwafemi Olukoya. Runtime and design time completeness checking of dangerous android app permissions against gdpr. *IEEE Access*, 2023.

[30] Christine Murad, Cosmin Munteanu, Benjamin R Cowan, and Leigh Clark. Revolution or evolution? speech interaction and hci design guidelines. *IEEE Pervasive Computing*, 18(2):33–45, 2019.

[31] Atsuko Natatsuka, Ryo Iijima, Takuya Watanabe, Mitsuaki Akiyama, Tetsuya Sakai, and Tatsuya Mori. 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*, pages 2633–2635, 2019.

[32] ABC News. 6-year-old mistakenly orders dollhouse, cookies worth $162 while chatting with amazon echo.

[33] Shidong Pan, Thong Hoang, Dawen Zhang, Zhenchang Xing, Xiwei Xu, Qinghua Lu, and Mark Staples. Toward the cure of privacy policy reading phobia: Automated generation of privacy nutrition labels from privacy policies. *arXiv preprint arXiv:2306.10923*, 2023.

[34] Shidong Pan, Zhen Tao, Thong Hoang, Dawen Zhang, Tianshi Li, Zhenchang Xing, Xiwei Xu, Mark Staples, Thierry Rakotoarivelo, and David Lo. A NEW HOPE: Contextual privacy policies for mobile applications and an approach toward automated generation. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 5699–5716, Philadelphia, PA, August 2024. USENIX Association.

[35] Shidong Pan, Dawen Zhang, Mark Staples, Zhenchang Xing, Jieshan Chen, Xiwei Xu, and Thong Hoang. Is it a trap? a large-scale empirical study and comprehensive assessment of online automated privacy policy generators for mobile apps. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 5681–5698, Philadelphia, PA, August 2024. USENIX Association.

[36] Google Play. Provide information for google play's data safety section.

[37] Alisha Pradhan, Kanika Mehta, and Leah Findlater. " accessibility came by accident" use of voice-controlled intelligent personal assistants by people with disabilities. In *Proceedings of the 2018 CHI Conference on human factors in computing systems*, pages 1–13, 2018.

[38] Muhammad Sajidur Rahman, Pirouz Naghavi, Blas Kojusner, Sadia Afroz, Byron Williams, Sara Rampazzi, and Vincent Bindschaedler. Permpress: Machine learning-based pipeline to evaluate permissions in app privacy policies. *IEEE Access*, 10:89248–89269, 2022.

[39] Aafaq Sabir, Evan Lafontaine, and Anupam Das. Hey alexa, who am i talking to?: Analyzing users' perception and awareness regarding third-party alexa skills. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–15, 2022.

[40] Irina Shklovski, Scott D Mainwaring, Halla Hrund Skúladóttir, and Höskuldur Borgthorsson. Leakiness and creepiness in app space: Perceptions of privacy and mobile app use. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2347–2356, 2014.

[41] Vijay Sivaraman, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, and Olivier Mehani. Network-level security and privacy control for smart-home iot devices. In *2015 IEEE 11th International conference on wireless and mobile computing, networking and communications (WiMob)*, pages 163–167. IEEE, 2015.

[42] George Terzopoulos and Maya Satratzemi. Voice assistants and smart speakers in everyday life and in education. *Informatics in Education*, 19(3):473–490, 2020.

[43] Matthias Urhahn. Permission pilot. https://github.com/d4rken-org/permission-pilot?tab=readme-ov-file, 2023.

[44] Sam Wolfson. Amazon's alexa recorded private conversation and sent it to random contact.

[45] George Wright. Amazon to pay $25m over child privacy violations.

[46] Fuman Xie, Yanjun Zhang, Chuan Yan, Suwan Li, Lei Bu, Kai Chen, Zi Huang, and Guangdong Bai. Scrutinizing privacy policy compliance of virtual personal assistant apps. In *Proceedings of the 37th IEEE/ACM international conference on automated software engineering*, pages 1–13, 2022.

[47] Chuan Yan, Fuman Xie, Mark Huasong Meng, Yanjun Zhang, and Guangdong Bai. On the quality of privacy policy documents of virtual personal assistant applications. *Proceedings on Privacy Enhancing Technologies*, 2024.

[48] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. 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. IEEE, 2019.

[49] Wei Zhang, Yan Meng, Yugeng Liu, Xiaokuan Zhang, Yinqian Zhang, and Haojin Zhu. Homonit: Monitoring smart home apps from encrypted traffic. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1074–1088, 2018.

[50] Yajin Zhou, Zhi Wang, Wu Zhou, and Xuxian Jiang. Hey, you, get off of my market: detecting malicious apps in official and alternative android markets. In *NDSS*, volume 25, pages 50–52, 2012.