Poster: A First Look at the Privacy Risks of Voice Assistant Apps

Atsuko Natatsuka
Waseda University
natatsuka@nsl.cs.waseda.ac.jp

Ryo Iijima
Waseda University / NICT
ryo@nsl.cs.waseda.ac.jp

Takuya Watanabe
NTT Secure Platform Laboratories
Waseda University
watanabe@nsl.cs.waseda.ac.jp

Mitsuaki Akiyama
NTT Secure Platform Laboratories
akiyama@ieee.org

Tetsuya Sakai
Waseda University
tetsuya@waseda.jp

Tatsuya Mori
Waseda University / NICT
RIKEN AIP
mori@nsl.cs.waseda.ac.jp

ABSTRACT

In this study, we conduct the first study on the analysis of voice assistant (VA) apps. We first collect the metadata of VA apps from the VA app directory and analyze them. Next, we call VA apps by the corresponding voice commands and examine how they identify users by analyzing the responses from the apps. We found that roughly half of the VA apps performed user identification by some means. We also found that several apps aim to acquire personal information such as birth date, age, or the blood type through voice conversations. As such data will be stored in the cloud, we need to have a mechanism to ensure that an end-user can check/control the data in a usable way.

KEYWORDS

voice assistance apps; measurement; privacy

ACM Reference Format:
Atsuko Natatsuka, Ryo Iijima, Takuya Watanabe, Mitsuaki Akiyama, Tetsuya Sakai, and Tatsuya Mori. 2019. Poster: A First Look at the Privacy Risks of Voice Assistant Apps. In *2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19), November 11–15, 2019, London, United Kingdom.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3319535.3363274

1 INTRODUCTION

Voice assistant (VA) systems execute various tasks and services by means of voice commands. The interaction between the user and a VA system is empowered by the VA app. In response to a user's voice request, the appropriate VA app is called through the VA system to process/answer the request. Typical voice assistant platforms, such as Google Assistant and Amazon Alexa, have a wide variety of VA apps that have been developed either officially or by third parties. VA apps are available in various platforms, such as AI speakers, smart TVs, and smart home systems.

The intrinsic characteristics of VA apps are that unlike apps for mobile platforms like Android and iOS, they run in the cloud and are *not* installed on devices. The developer of a VA app deploys the program of the VA app on an official platform or their own server. The problem here is that the behavior of VA apps is not transparent to users. That is, the use of the information obtained by the VA app through the conversation with the user is determined by the program running in the cloud, which the user cannot directly access, and there is no way for the user to know where the information is stored or how the information is used. In fact, the user study by Abdi et al. [1] clarified that the awareness of security and privacy in voice assistant use is weak and that a majority of users are not aware of the existence of a third party.

Given the privacy risks of VA apps, discussed above, we are conducting a large-scale measurement study of VA apps. We adopt Google Assistant as a representative voice assistant system and analyze the VA apps available on the official marketplace, Assistant Directory. We address the following research questions (**RQs**)

RQ1: How many VA apps exist to identify users and how do they identify users?
RQ2: How many VA apps access personal information, and what kind of information do they obtain?

The key technical challenge of this study is analyzing the behavior of VA apps running in the cloud. VA apps do not run on the device, so application analysis methods such as static and dynamic analysis cannot be used. In order to analyze the behavior of the VA app, we need to analyze the *dialogue*; i.e., we need to analyze the voice commands that are input and the responses output by the app, implying that the analysis should rely on natural language processing[1].

In order to address the research questions, the metadata of 940 of the available VA apps from the Assistant directory [3] must first be collected. Next, using the extracted voice commands from the obtained metadata, the corresponding VA apps will be activated to start the conversation. Using the information obtained through the conversation with the VA apps, the VA apps will be classified into several classes, based on the techniques used for identifying users. The contributions of this study include the following:

- It is the first extensive study to be conducted on the VA apps in the wild.

[1] In voice assistant systems, all the speech data is converted to text data.

CCS '19, November 11–15, 2019, London, United Kingdom
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6747-9/19/11.
https://doi.org/10.1145/3319535.3363274

Table 1: User Identification Schemes

Scheme	Require user approval?
OAuth (account linking)	Yes
Google Sign-In (account linking)	Yes
Helpers Intent	Yes
UserStorage	No
userId (deprecated)	No
lastSeen	No

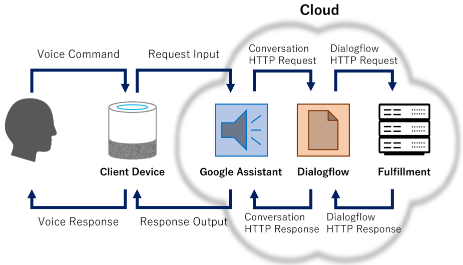

Figure 1: High-level Overview of VA apps (Actions on Google).

- A framework has been developed to classify the user identification techniques used by VA apps.
- It reports the typical characteristics of VA apps, the user identification techniques, as well as the privacy risks.

2 BACKGROUND

Figure 1 presents the high-level overview of Actions on Google [2], which is a development platform for the Google Assistant. Google Assistant is an interface of the VA application, and it converts the input request by the voice, into the natural language, using the voice recognition system, and transmits the Conversation HTTP request to Dialogflow. The Conversation HTTP request contains several additional pieces of information, such as user information and user device information. Dialogflow is a natural language processing tool that analyzes the user's text input, extracts words necessary for starting and processing the VA application from the received request text, and sends them to Fulfillment in a Dialogflow HTTP request. Fulfillment is a VA app in the cloud that creates a response to be displayed to the user from the extracted words. The generated response will be sent back to the user through Dialogflow, Google Assistant, and the client device. Note that while Google Assistant and Dialogflow run on the platform offered by Google, Fulfillment may run on the platform offered by Google (i.e., Firebase) or may run on the other platforms provided by the third party, in which case only developers can fully understand/monitor the behavior of apps. From the user's point of view, it is difficult to understand what kind of data is transmitted and received in the cloud. Table 1 summarizes the user identification schemes available on Google Assistant. Due to the space limitation, we omit the details of each scheme. However, interested readers can refer to the official documents [2]. While the first three—two account-linking schemes and Helpers Intent—require user approval in advance, the last three—UserStorage, userId, and lastSeen—do not require user approval; hence, the use of these schemes could be opaque to end-users. We note that of the three latter schemes, userId, which represents anonymous user identify, has been deprecated since July 29, 2019.

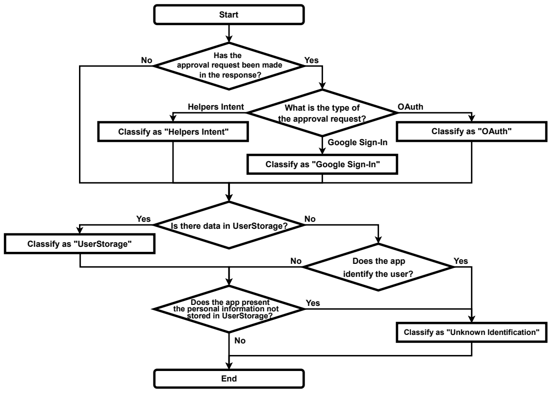
Start
Has the approval request been made in the response?
No
Yes
What is the type of the approval request?
Helpers Intent
OAuth
Google Sign-In
Classify as "Helpers Intent"
Classify as "OAuth"
Classify as "Google Sign-In"
Is there data in UserStorage?
Yes
No
Classify as "UserStorage"
Does the app identify the user?
No
Yes
Does the app present the personal information not stored in UserStorage?
Yes
No
Classify as "Unknown Identification"
End

Figure 2: A Flowchart of Classifying User Identification Schemes.

3 MEASUREMENT

3.1 Collecting Metadata

We first collect metadata from the Assistant directory, which can be seen as a marketplace of Google Assistant apps operated by Google. The directory provides an interface to explore all the available Google Assistant apps in a region. Each app page contains the app name, developer name, description, voice commands to activate the app, categories, supported devices, average rating, review, privacy policy, and UserStorage data. Using the data crawler that we developed, we collected the metadata of 940 apps that were released in the Japan region on July 7, 2019. Of the 940 apps, we extracted 782 apps that were developed by third-party developers.

3.2 Collecting App Behavior

Next, we collect app behavior by calling apps through the corresponding voice commands. To this end, we use the Google Assistant SDK and Android Debug Bridge. This approach enables us to input the voice commands in the form of plain text; thus, we can easily automate the process of collecting app behavior. In this work, after calling the VA app and obtaining the first response, we terminate the VA app without sending a further request. To create the appropriate second request, we need to analyze the context of the conversation. Achieving such a task is not straightforward; we leave it for our future work. After we call all VA apps, we check the corresponding UserStorage. To determine whether an app was able to identify a user, we repeat the same process twice as some apps generate a different response in the second run, e.g., "Hello again, Alice," implying the success of user identification.

3.3 Analyzing the User Identification Schemes

Figure 2 shows a framework for analyzing the user identification schemes of a VA app. First, a response of a VA app is analyzed to verify whether the app sends a request for the approval of account linking or Helpers Intent. If we find that an app has made an approval request, the app is classified into one of the following groups: *OAuth*, *Google Sign-In*, or *Helpers Intent*, which are distinguishable by checking the template of response. Next, we check for the presence of data in *UserStorage*. If there is non-empty data

Table 2: Breakdown of VA apps according to their user identification schemes. As some apps used multiple schemes, the total number is not equal to the summation of the apps.

Descriptions	#apps
No user identification	371
Account linking	45
Helpers Intent	10
UserStorage	310
Unknown	42
Not callable	16
Total	782

Table 3: Statistics of the Personal Information Aqcuired by VA Apps.

Information	Method	#apps
Birth date	Conversation	16
Name	Conversation	2
	Google Sign-In	10
	Helpers Intent	1
E-Mail address	Conversation	1
	Google Sign-In	10
Profile photo	Google Sign-In	10
Present location	Helpers Intent	10
Address	Conversation	7
Age	Conversation	2
Blood type	Conversation	2
Phone number	Conversation	1
Gender	Conversation	1

in *UserStorage*, the user identification scheme of the corresponding app is classified as the use of *UserStorage*. If there is no data in *UserStorage*, we compare the first response against the second response.If the second response has been changed from the first response and contains some personal information such as names or locations, the app successfully identifies the user. If we find an app that identifies the user without using the schemes shown above, the app likely uses *userId*, *lastSeen*, or other schemes we do not know; these schemes are not distinguishable by nature because the use of these schemes is invisible to end-users. We classify such cases as *Unknown Identification*. We also check whether or not *UserStorage* is used for storing information about the user.

4 VA APPS

Table 1 shows the breakdown of VA apps according to their user identification schemes. We find that of the 782 apps, 395 (= 782 − 371 − 16) apps used at least one of the four identification schemes shown in the table. In other words, roughly half of the apps that we studied performed user identification by some means or another. *UserStorage* was the most popular identification scheme used for identifying users. We noted that it was not easy for a typical end-user to check/control the content of UserStorage, as it required searching for the app-specific web page on the Assistant directory and accessing the page by a PC browser. Table 3 presents the statistics of the personal information acquired by VA apps. While it is not surprising that the officially supported schemes, such as the account linking (Google Sign-In,) which requires the personal information, such as a person's name, e-mail address, or profile photos, it was somewhat surprising that many of VA apps attempted to collect personal information through their conversation, i.e., the user is asked questions, such as "When is your birthday?" The birthday information was used by a fortune telling app. As the data obtained by the VA apps could be stored on the third-party fulfillment server, keeping track of the way in which the obtained data is used/distributed is not feasible.

5 DISCUSSION

This work has two limitations. Firstly, we targeted the VA apps of Google Assistant released in Japan, and we have not studied the apps released in other regions. Expanding our analysis to other VA apps, including other regions and other platforms such as Amazon Alexa, is left for future work. Second, in this work, we only analyzed the first response made by a VA app. By continuing the conversation with the VA app, we can expect to analyze the behavior of VA apps regarding access to privacy-sensitive information through the conversation. In our future work, we will extend our analysis to achieve this goal.

6 RELATED WORK

Abdi et al. [1] conducted a survey on user perceptions of VAs and found that user perceptions of the VA and related data activities were incomplete. In addition, they verified the VA threat model and proposed a non-technical strategy to protect the user from attacks. Furthermore, they proposed a method to meet user security and privacy expectations for a VA. Zhang et al. [4] proved, by conducting a user study and from actual attacks, that there is an attack method called a voice squatting attack (VSA), hijacking voice commands of VA apps, and an attack called a voice masquerading attack (VMA), impersonating the VA or the VA app. Also, they suggested the possibility of VSA being performed by using a scanner that finds similar VA app names. Besides, they designed a situation-dependent detector that identifies malicious apps performing VMA, and it achieved 95% accuracy.

7 SUMMARY

This work is the first to employ an extensive analysis of VA apps in the wild. Our analyses revealed that of the 782 VA apps we analyzed, at least 395 employed user identification by some means. Of the apps with user identification, 310 of them made use of UserStorage as a means to identify users. As it is not straightforward for a user to check/control the content of UserStorage, we need a usable mechanism to make the data more transparent to users. In this study, we take a look at the beginning of the conversation between a user and a VA app, limiting our analysis to a part of the entire conversation. In our future work, we will extend our conversation analysis by addressing the *context* of the conversation with the natural language spoken dialogue system; such analysis will be useful to automate testing the behavior of VA apps.

ACKNOWLEDGMENTS

A part of this work was supported by JSPS Grant-in-Aid for Scientific Research, Grant Number 18K19789 and 19H04111.

REFERENCES

[1] Noura Abdi, Kopo M. Ramokapane, and Jose M. Such. 2019. More than Smart Speakers: Security and Privacy Perceptions of Smart Home Personal Assistants. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 451–466.

[2] Google. 2019. *Actions on Google*. Google.

[3] Google. 2019. *Assistant directory*. Google.

[4] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. 2019. Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems. In *Proc. of the 2019 IEEE Symposium on Security and Privacy*. IEEE, 263–278.