Alexa, is the skill always safe? Uncover Lenient Skill Vetting Process and Protect User Privacy at Run Time

Tu Le[*]
tul10@uci.edu
University of California, Irvine
Irvine, CA, USA

Dongfang Zhao[*]
zhaodo@iu.edu
Indiana University Bloomington
Bloomington, IN, USA

Zihao Wang
zwa2@iu.edu
Indiana University Bloomington
Bloomington, IN, USA

XiaoFeng Wang
xw7@indiana.edu
Indiana University Bloomington
Bloomington, IN, USA

Yuan Tian
yuant@ucla.edu
University of California, Los Angeles
Los Angeles, CA, USA

ABSTRACT

Voice personal assistant (VPA) platforms (e.g., Amazon Alexa) allow developers to deploy their voice apps on third-party servers. However, this strategy introduces unexpected privacy risks to VPA customers. Malicious developers can dynamically change their app's behaviors to circumvent the platform's vetting process. This paper aims to systematically analyze Alexa's voice app ecosystem (i.e., Alexa skills), focusing on behavior manipulation (also referred to as skill behavior change). We identify the root causes of malicious skills getting published and propose a defense solution to effectively protect users. First, we uncover Amazon's skill vetting strategy and the privacy issues relevant to their vetting. We reveal that, in addition to the skill certification process before a skill gets published, Amazon also deploys a skill monitoring scheme after the skill is published. We further discover limitations of this monitoring scheme that have not been explored in previous research. Lastly, to address these issues, we propose a run-time skill monitoring approach to check the consistency of the skill behaviors when users interact with skills. Our findings suggest a call for action to improve the vetting process for VPA skills without placing a burden on skill developers and help developers adhere to policies.

CCS CONCEPTS

• **Security and privacy → Domain-specific security and privacy architectures**; **Privacy protections**; **Usability in security and privacy**; • **Human-centered computing → Personal digital assistants**.

KEYWORDS

voice application, privacy, runtime, monitoring, vetting, Alexa

ACM Reference Format:
Tu Le, Dongfang Zhao, Zihao Wang, XiaoFeng Wang, and Yuan Tian. 2024. Alexa, is the skill always safe? Uncover Lenient Skill Vetting Process and Protect User Privacy at Run Time. In *Software Engineering in Society (ICSE-SEIS'24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3639475.3640102

LAY ABSTRACT

Our homes are getting smarter with voice-controlled devices like Amazon Alexa offering automation and convenience. By adding voice applications (also called "skills"), which could be hosted by third parties, the abilities of the devices keep expanding. However, our research reveals the dark side of this technology - possible privacy breaches. Malicious developers can modify the behaviors of their skills after bypassing Amazon's scrutiny, potentially violating user privacy. Although Amazon has preventive measures in place (e.g., skill certification and repeated monitoring), adversaries can still slip through the cracks. In this paper, we comprehensively uncover the loopholes with real-world examples for better understanding and propose a solution to effectively help users monitor dynamic skill behaviors.

1 INTRODUCTION

The global intelligent virtual assistant market was valued at billions of dollars and is expected to continuously grow from 2021 to 2028 [28]. As smart software agents that can provide services based on user commands or questions, virtual personal assistants (VPA) like Amazon Alexa and Google Assistant are becoming increasingly popular with families around the world.

In this paper, we perform a systematic study of the Alexa skill ecosystem, especially the risks caused by the developers' ability to dynamically change a skill's back-end code. We find that the vetting policy enforced by Amazon follows different criteria from what is claimed in developer requirements. We uncover the skill monitoring process that has not been studied before. Besides, Amazon allows developers to deploy a skill's back-end logic to a third-party server that is not under Amazon's control. This is very risky because malicious developers can submit a benign skill and change its back-end behavior logic after the skill is published in the skill store. In this way, it is able to circumvent the whole vetting process and permission model easily. A skill verified as a benign one may request user personal information (e.g., phone number) verbally. We call this risk *skill's behavior change.* When a skill's behavior change happens, users may be credulous and just reveal personal information. We also find some evidence showing such malicious

[*] contributed equally to this research.

ICSE-SEIS'24, April 14–20, 2024, Lisbon, Portugal
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0499-4/24/04
https://doi.org/10.1145/3639475.3640102

skills do exist in the wild. Even though Amazon strengthened skill vetting by appending a new monitoring process to the initial vetting process, we find it is still insufficient to defend against a skill back-end change. Our test skills successfully pass the vetting process and perform malicious behavior changes without being detected. To help protect user privacy, we propose an effective real-time skill behavior monitoring system that detects suspicious skills and notifies the user during interactions.

Skills deployed on external servers can introduce privacy risks to users [23] because malicious developers can change their back-end code anytime after skill approvals. However, there is no prior research about skill behavior monitoring approaches to detect malicious behavior changes, mainly due to the following challenges.

Challenges. First, the skill response, or the *behavior* of skill, should always be consistent with its functionalities. When focusing on the suspicious skill response asking for personal information, we need to distinguish the information request (e.g., "What's your phone number") from other responses (e.g., "Do you want to continue?"). This needs a corresponding natural language processing method. In addition, the most challenging part is to analyze the skill description, which describes the skill functionalities. The variability of semantic expression makes skill descriptions diverse even when two skills are doing the same thing. Straightforward keyword-based approaches can only handle those skill descriptions with matched keywords. It is a challenge to develop a semantic interpretation approach to handle the ambiguity of natural language and find the correlation between skill response and skill description. Second, when developers put the skill's source code in an external server, the skill behavior logic (how the skill responds to user utterances) becomes a black box. Moreover, malicious developers can change the back-end code anytime to request private information from users. This issue can only be solved by a real-time skill behavior monitoring system that runs in the background whenever the user communicates with skills. However, a real-time running system must have an unnoticeable delay, which requires the skill monitoring system to have high accuracy and low latency.

Key Contributions. This paper has the following contributions:

- We discover the monitoring process after a skill gets published and reveal the criteria Amazon follows to identify a violating skill, which has not been studied before (Section 3.1 and Section 3.3).
- We discover a new attack vector called "versatile intent" that allows adversaries to stealthily collect any type of information by manipulating their published legitimate skill (Section 3.2).
- We identify the necessity for a defense system at run-time. In particular, the service provider does not have full control over the published skills, especially those hosted on third-party servers. The behavior changes of such skills are unpredictable. Our experiments show that the skills could adopt certain patterns for their behaviors to bypass the vetting (Section 3.4).
- We propose a run-time monitoring solution when the user is interacting with the skills (Section 4). The source code of our core techniques will be released for future research. We then show that privacy-invasive skills are still an underlying threat (Section 4.5). Of 637 skills that request personal information from users in our dataset, there are 141 suspicious skills without proper permissions or functionalities.

2 RELATED WORK

Virtual personal assistants (VPA) have been studied as one category among general IoT devices [13, 15, 32]. But lately, VPAs have drawn attention as a popular category of IoT equipment. Many researchers put effort into how to attack or fool the system through adversarial attacks against speech recognition [9–11, 35]. Some researchers focused on defenses against such attacks [2, 25].

The speech recognition system, which is the core of the voice assistants, was known to have misinterpretation vulnerability [7, 19, 29, 36]. Exploiting the misinterpretation problem, an adversary could impersonate the voice assistant system or other skills to eavesdrop on users. Researchers found approaches for malicious Alexa skills [8] to eavesdrop on users' conversations. Apart from the speech recognition component, prior work [37] also analyzed and evaluated the security of the succeeding Natural Language Understanding (NLU) component.

Smart speakers are a black box to users. Based on some previous studies [1, 17], users have an incomplete understanding of the smart speaker model and are concerned about their privacy [20]. Recently, researchers have put more effort into reverse-engineering the smart speakers' vetting mechanism [12, 31]. Cheng et al. [12] built plenty of test skills to disclose the skill certification process.

Comparison with prior work: Zhang et al. [37] only discussed how the developers replace back-end audio files. Lentzsch et al. [23] illustrated how to exploit dormant intents to manipulate the back-end code logic of the skills. In our work, we study the back-end code change risks more thoroughly with more test skills. In particular, we discover a new approach called "versatile intent". Previous studies [12, 18, 34] identified policy-violating skills and found that Amazon conducted a vetting process differently from what is described in its documentation. However, they did not explore the behind-the-scene vetting criteria. We detail the criteria in Section 3.1. Besides, we uncover that there is not only an initial skill certification process but also a monitoring process after the skills get published. We investigate the whole vetting pipeline to provide a more comprehensive understanding of how Amazon manages the skill ecosystem (Section 3.1 & Section 3.3). Liao et al. [24] analyzed the consistency between the privacy policies provided by the skills and the corresponding skill descriptions. However, skill description might not cover everything the skill actually does and privacy policy is not a reliable enforcement. In contrast, we focus on the actual behaviors of the skills at run-time. While prior work from Guo et al. [18] performed skill description and skill behavior analysis, they employed a manual analysis approach on 100 selected skills. In our work, we identify the need to protect users at run-time and present an automated approach, which is the foundation of our run-time skill monitoring system (Section 4).

3 SYSTEMATIC STUDY OF ALEXA SKILL VETTING AND PRIVACY RISKS

First, we uncover Amazon's criteria for vetting (Section 3.1). Second, previous research mostly studied the skill certification process that happens before the skill gets published [12, 31]. We discover that the skill certification process is just the first stage of the whole vetting pipeline. We take a further step to uncover the skill monitoring process happening after a skill gets published (Section 3.3). We also demonstrate approaches that can bypass the whole vetting pipeline to perform malicious skill behavior changes (Section 3.4).

3.1 Demystifying Amazon's Skill Vetting

Finding I: Previous studies found skills violating privacy policy on the market or asking for personal information without declaring permissions [12, 18, 24], which was concluded to be the limitation of Amazon's vetting. We further uncover how Amazon's vetting actually works to better understand why such violating skills were allowed to be published. In short, it follows a criterion: functionality correlation. The user information requested by skill is not necessarily declared in the permission list. The skill can ask for the user's personal information through voice interaction as long as it declares such information is necessary for the skill's functionalities.

Alexa developer documentation regulates how the skills request user information. In summary, it requires the developer to provide a permission list and describe the skill's functionalities to the users if the skill requests any kind of personal information. Although the documented vetting policies have detailed and strict requirements, the actual vetting process works differently. For example, many published skills did not provide a privacy policy when they were supposed to. Prior studies [12, 24] presented such measurements. However, details about how Amazon's vetting process actually works are still underexplored.

3.1.1 *How Amazon enforces the policies.*

Cheng et al. [12] showed that some skills asking for a user name in the first interaction could still pass the certification process even though they did not claim any permission. If Amazon followed the vetting criteria precisely the same as its documentation, they would not have missed the violations in the first interaction with the malicious skills. What are the tricks for these skills to bypass Amazon's vetting? It is very likely that Amazon is enforcing vetting criteria differently from what is claimed in the documentation. Previous works attributed it to Amazon's leniency, which oversimplified the issue. We uncovered and validated that "leniency" follows some rules. To explore this issue, we built some test skills to reverse engineer the vetting process.

If a skill requests personal information, the skill needs to provide the following three items according to the developer documentation: the description to depict skill functionalities, a permission list, and a privacy policy disclosing the requested information. We designed some test skills to make it clear which parts are not considered seriously by Alexa vetting through the controlled variable method. We first built our test skill *Mascot Box* without a permission list (it still had a privacy policy and skill description). The skill's functionality is to provide some sweet words to the user when invoked. However, it will also ask for the user's phone number in the first response. As a result, it failed the certification process of which Amazon's feedback said *"... Your skill is requesting information that is not relevant to the skill's functionality. Namely, your skill request phone number."* The feedback inspired us to think that an important vetting criterion could be the correlation between functionality and actual behavior. So we validated this criterion by re-making a test skill to first ask for *full name* and then give a greeting response "Hello, *A*" after the user provides his/her name *A*. This time, the test skill successfully passed the certification process. To further confirm the discovered criterion for other types of personal information, we built similar test skills that asked for a phone number, email, and location. Test skills include test skill *Sweet Text* which will send a sweet message to the user's phone number or email, test skill *My Weather* which provides local weather to the user, etc. They all became qualified skills and were published in the store.

The above experiment shows that the skills are allowed to collect customer contact information if the request information has a correlation with the skill functionalities, which can explain why some aforementioned test skills violating privacy requirements could still pass the certification process in prior works. For example, the test game skills [12] asking for a name in the first response passed the certification process because the Alexa vetting team considered a user name could improve user experience (i.e., it has a correlation with the skill functionality to some degree). Specifically, regarding *name* information, we found any game skills were allowed to ask *"What's your name?"* as long as a greeting *"Hello, [name X]"* followed after the user provided a name.

3.2 Manipulating Skill Behaviors at Run-time

Finding II: Malicious developers can change the back-end logic of skills deployed on the third-party server after the skills get published on the store to have users disclose personal information. The back-end change can be achieved by dormant intents and "versatile" intents. The latter approach is novel and stealthier, motivating the need for a run-time monitoring approach while the user is interacting with the skills.

When a skill is deployed on a third-party server where Amazon can not access its back-end code, developers can arbitrarily change the skill's behavior logic at any time. This makes the skill able to dynamically change its behavior after it gets published on the store, e.g., the skill can ask for a user phone number that has no correlation with its functionalities. Credulous users may just reveal their information when they are requested. Currently, there is no vetting mechanism that is able to detect such suspicious behavior changes.

Malicious developers have two approaches to exploit the skill's behavior changes for their published skills. The first approach is to leverage "dormant" intents, which was previously discussed by Lentzsch et al. [23]. In our work, we discovered a new approach that leveraged "versatile" intents. We describe the two approaches and provide the comparisons as follows.

3.2.1 *Dormant intent.*

A malicious developer can craft a safe skill with an unused intent (so-called "dormant intent") that collects certain sensitive information (e.g., phone number). The vetting process will not find any suspicious behavior of the submitted skill because the trigger logic does not get designed for the dormant intent yet. However, after the skill gets published, the developer can change the back-end code to activate the dormant intent. Whenever users trigger that intent, the skill will ask the users for the sensitive information that the intent was crafted for. Figure 1 shows how a skill that originally does not request any personal information gets its dormant intent activated after passing the certification process.

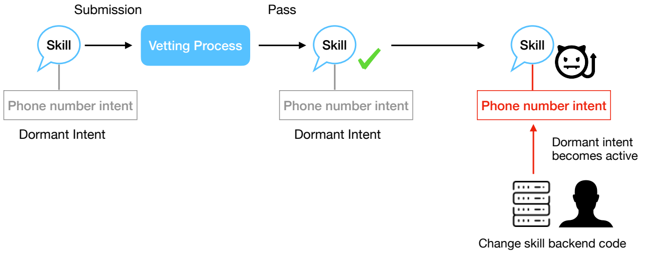


Figure 1: Overview of how a "safe" skill passes the vetting process then activates its dormant intent for the usage of collecting sensitive information

3.2.2 ***Versatile intent***. Amazon maintains a list of slot types that define how phrases in utterances are recognized and handled for intents, such as *AMAZON.PhoneNumber, AMAZON.City, AMAZON.Color,* etc. No matter which kind of information a skill wants to obtain on the back end, it has to choose a corresponding intent slot type for its purpose. There are two categories of slot types: *List* and *Numbers/Dates/Times*. *List* includes slot types that represent a list of items (text), while *Numbers/Dates/Times* includes slot types that convert the user's utterance into data types such as numbers and dates. Amazon tried to provide separate slot types for different types of information to prevent developers from requesting more than what they need [1]. However, malicious developers can use an intent designed for information *A* to collect information *B*; we call such intent "versatile intent". We found this problem from developing our test skill, named *"Guess Number"*, which is a game skill that generates a random integer number and asks the user to guess it. The *AMAZON.Number* intent is necessary for its basic functionality, but a malicious developer can secretly leverage it to collect personal information such as phone numbers through back-end change. Similarly, there are many other versatile intent slot types that can be exploited. For example, *City* types can be used to collect user names. Interestingly, we identified several versatile intent slot types that are non-restricted (Table 1). Different from regular slot types, these slot types can record any text or number inputs. For example, *AMAZON.StreetName* can be used to collect any other information such as email address.

3.2.3 ***Versatile vs Dormant***. Versatile intent exploit has not been discussed before. Versatile intents are more concealed loopholes compared to dormant intents. Dormant intents are unused intents that could be easily detected during the vetting phase. Amazon can check all intents crafted in the skill and identify unused intents that can be dormant intents. However, it is different for versatile intents. Versatile intents are intents originally used for legitimate skill functionalities (satisfying vetting requirements) but can be exploited for a different purpose at run-time. An adversary can build a skill that necessarily has certain versatile intents serving its functionality and then later exploit them to collect different personal information. There is no way for Amazon to foresee whether or when the skill's back-end logic would be modified for malicious purposes. In this case, run-time monitoring while the user is interacting with a skill is needed.

Table 1: Non-restricted intent slot types provided by Amazon Alexa. These slot types have blank descriptions and can be exploited to collect any information.

Category	Non-restricted Slot Types
List (Text)	AMAZON.Anaphor AMAZON.RelativePosition AMAZON.StreetName AMAZON.VisualModeTrigger
Numbers/Dates/Times	AMAZON.Ordinal AMAZON.PhoneNumber

3.3 Reverse Engineering Amazon's Skill Monitoring Process

Finding III: The so-called monitoring process is an underexplored process in the skill vetting mechanism. It intermittently tests the behavior of recently certified skills in the store during a certain period. Suspicious skill behavior will cause skill removal. However, it is a periodic testing process instead of a supposedly continuous monitoring process and can be bypassed easily because it follows certain patterns.

Prior studies [12, 23] put a lot of effort into figuring out how Amazon's skill certification process works and its loopholes. However, that is not the whole picture of the vetting process. Even though a skill passes the certification process, the vetting is not finished yet. Amazon repeatedly tests published skills for a period, which is called the "monitoring process". If the skill shows suspicious behavior, i.e., asking for personal information which has no correlation with its functionalities, it will get removed from the store. When our test skill passed the certification process, we changed its back-end logic to ask for personal information unreasonably. After several days, we received feedback informing us that the skill did not pass the "monitoring process" and was removed from the store. To the best of our knowledge, this monitoring process has not been studied before.

To understand how this monitoring process actually works, we built and submitted multiple test skills of different functionalities to the Alexa platform. On the skill back end, we logged the requests made to each skill. We illustrate our analysis and findings in the following paragraphs of this section.

3.3.1 ***Misleadingly named "monitoring process"***. Amazon claimed that they deployed a monitoring process that continuously monitors and tests the skills. To understand how this monitoring process works, we first published three safe skills. Once they got published in the store, we changed the back-end logic of our skills to ask for personal information unreasonably and not disclose it to the user. All those three skills were not removed from the store

[1] https://developer.amazon.com/en-US/docs/alexa/custom-skills/slot-type-reference.html

until seven days later. Our experiment suggests that Amazon's monitoring is actually not a continuous process, which could leave an attack window for around seven days after the skill gets published.

3.3.2 ***Functionality consistency-based monitoring.*** As illustrated in Section 3.1, whether the information request from the skill has a correlation with its functionalities determines if the skill is able to pass the Amazon vetting mechanism. With our test skills, we validated that the criterion is a generic standard applied to all information permissions, including name, postal address, phone number, email, and zip code.

We first had five skills published in the store, then changed their back-end code to request those above five different types of personal information without reasons and corresponding permission requests. The skills were intentionally modified to ask for user information in the first response whenever launched so that Amazon's monitoring process would not miss these suspicious requests. All skills were removed from the store within around one week, just like the previously published three test skills. Next, we published another five test skills, and again, we deliberately changed the back-end logic to have them ask for the five kinds of information, respectively. However, the different part in this round was that those information requests have a correlation with the skill's functionalities. As a result, 5/5 skills survived the monitoring process this time even though they did not have corresponding permissions. This experiment validated that the vetting criterion for suspicious skill determination is based on whether the requested information is consistent with the skill functionalities.

3.3.3 ***Periodic monitoring pattern.*** Since we found that Amazon's skill monitoring is not a continuous process, we wanted to further discern the pattern. To do this, we collected backend logs from our test skills, which were subjected to Amazon's test queries. However, Amazon anonymizes these test requests, making them indistinguishable from genuine user interactions. The concurrent use of our test skills by Amazon's vetting process and potentially real users complicated the process of identifying Amazon's monitoring patterns. To address this challenge, we undertook a retrospective examination of the activity logs from our test skills. We deduced Amazon's request pattern based on the types of requests made and the timing of our skill deactivation. Figure 2 shows the skill activity visualization metrics of our three test skills published on the same day. The first round of requests was logged on November 20th, approximately seven days post-publication, signifying the commencement of Amazon's monitoring phase. Our test skills, published on varying dates, exhibited a similar pattern of incoming requests, as emphasized in Figure 3. This strongly suggests that Amazon's monitoring process was predominantly active on workdays, particularly Monday, Tuesday, and Wednesday. In figure 3, the depicted color gradient signifies the aggregate number of unique request types directed at our test skills. A higher diversity of request types indicate the presence of Amazon vetting, given that typical users seldom utilize certain request types like FallbackIntent and NavigateHomeIntent, favoring more intuitive ones such as YesIntent and NoIntent. If the diversity of request types is four or fewer, it likely signifies consumer-generated requests. Conversely, a diversity exceeding four request types could suggest the influence of Amazon vetting within that day's requests. Note that our test skills with obviously suspicious behaviors were also detected and taken down on such days after those types of requests happened. In conclusion, our inference suggests that (1) the monitoring process spans over a period of seven weeks, and (2) the testing predominantly takes place on workdays (especially Monday, Tuesday and Wednesday), based on Eastern Standard Time (EST). Given the discernible pattern in the monitoring process, it provides ample opportunity for an attacker to strategize bypass mechanisms. This could involve an initial phase of inference-based attacks to discern the monitoring pattern, followed by crafting of specialized attacks in response. We have carried out a series of proof-of-concept experiments to validate this idea.

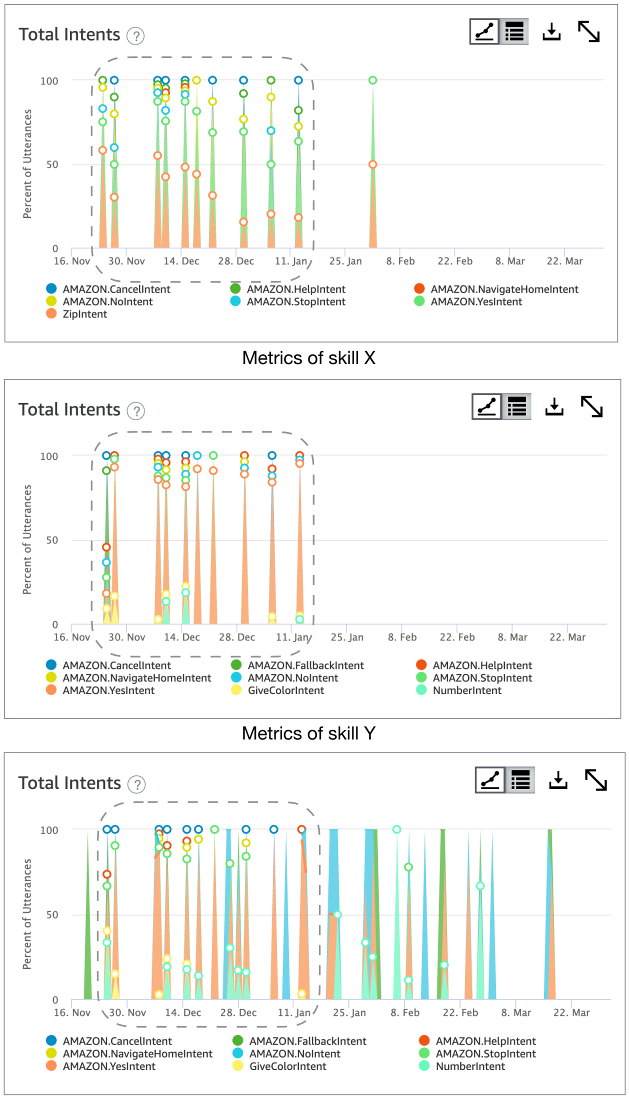
3.4 Bypassing Amazon's Skill Monitoring

Finding IV: We identified two approaches that can be used to bypass Amazon's skill monitoring. These approaches allow a skill to collect personal info that is not needed for its functionalities.

We published ten test skills and then changed the back-end logic to test how suspicious developers can circumvent the monitoring process. We tried different tricks to avoid being detected by the monitoring process. Not all test skills worked to evade the vetting. For example, we tried probability-based approaches, which means we set suspicious information requests that may occur in a chance of 1/20 or 1/30. We also tried an approach based on the number of interactions which means we set the suspicious request to occur after 5 or 10 customer interactions. These two methods did not survive the monitoring process. By checking the back-end logging of the four corresponding skills, we found Amazon would check a common intent ten times, e.g., YesIntent, which triggered our aforementioned trick settings. It is good to know that Amazon is trying to perform comprehensive vetting by checking a single intent many times. However, malicious developers are still able to easily evade the vetting by the following methods. We found the following two approaches that can bypass the monitoring process:

(1) Time-based approach: The skill only collects user info in a certain period of time. It is easy for malicious developers to figure out the periodicity of the Amazon monitoring process by logging the coming requests on the back end. They can wait until the monitoring process is finished before making malicious back-end changes that request personal information. We published a time bomb test skill that would ask for a phone number from 14:00 to 16:00 (EST) each day after the Amazon monitoring process is finished. Our skill was alive on the store for several months without being detected by Amazon vetting until we removed it, which proves the feasibility of this attack approach.

(2) Pattern-based approach: The skill only collects user information when the user interacts with it following a certain intent pattern. When a skill has multiple intents, it naturally has many different intent invocation paths of which Amazon did not yet try to cover all the possibilities. For example, our published test skill *"Lucky food"* asked for the user's phone number only when the user gave the following responses to the skill in order: "give me a food, give me a food, yes, no." It also survived the monitoring process.



Metrics of skill X

Metrics of skill Y

Metrics of skill Z

Figure 2: Metrics of skill activities of three test skills after they pass certification and get live. It is inferred that Amazon's monitoring process occurs during the highlighted period. Utterances outside this highlighted span are likely attributed to regular users, as these interactions exhibit greater randomness and predominantly trigger intuitive intents(e.g. Yes/No intents).

4 RUN-TIME SKILL MONITORING

We first discuss the threat model (Section 4.1). We then describe the design, workflow, and validation of our system, including two main components: Skill Behavior Check and Run-time Protection (Sections 4.2, 4.3, and 4.4). Finally, we present our measurement study and usability testing (Sections 4.5 and 4.6).

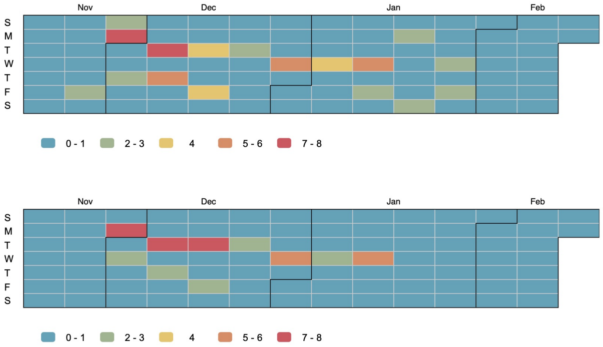
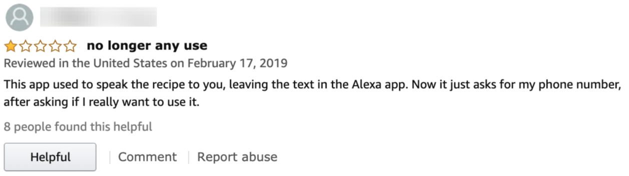


Figure 3: Different request types received by test skills which got live on different days. Here, the rows denote the seven days of the week, while each column signifies a consecutive week. The color intensity indicates the diversity of request types received. Notably, workdays (especially Monday, Tuesday, and Wednesday) register the highest variety of requests, suggesting Amazon's monitoring activity primarily transpires on these days, a pattern consistent across approximately seven weeks.

4.1 Threat Model

We consider a threat model where an adversary intentionally develops malicious skills to collect personal information from users. As demonstrated in the previous section, Amazon's vetting could fail to detect such malicious skills with behavior changes or bypassing techniques. Hence, users have to be aware of what information they give to the skills to protect their privacy. However, humans often make mistakes, and no installation or download when invoking a skill makes it even easier for malicious skills to bypass user awareness. Figure 4 shows a negative customer review about a skill that used to work fine but now changes its behavior to suspiciously ask for the user's phone number. It is an underlying issue that most users might not be aware of the risks.


[redacted]

★☆☆☆☆ **no longer any use**

Reviewed in the United States on February 17, 2019

This app used to speak the recipe to you, leaving the text in the Alexa app. Now it just asks for my phone number, after asking if I really want to use it.

8 people found this helpful

Figure 4: Negative review of the skill "The Bartender" which used to function properly but later changed its behavior to suspiciously collect user information.

4.2 System Overview

Skill behavior change can be a threat to user privacy if exploited by malicious developers. The back-end code of a skill deployed on a third-party server is out of Amazon's control. Amazon is enforcing an extra monitoring process after skill approval. However, it is hard to detect a malicious skill as it is unpredictable when/how the skill may ask for personal information. Thus, we propose to apply a run-time monitoring solution.

Our system determines a skill is suspicious if the information request is inconsistent with the skill's functionalities. In section 3 we show that Amazon is actually enforcing functionality-based vetting criteria. Therefore, we are following the same criteria to build our monitoring system, focusing on the skill's description. Besides, we consider the skill safe when it has a corresponding permission list for the requested information. That is because the skill with a permission list needs a user's explicit grant by either utterance or clicking a mobile prompt. Note that we will not take the skill's privacy policy into consideration. Because for many skills, the privacy policies are too general, often over-claiming the information that may be used, making us choose not to use privacy policy content as a vetting criterion.

Our system runs in the background whenever users interact with skills, which will notify users of potential privacy risks (e.g., when a skill is asking for a user's personal information, which does not correlate with skill functionalities). Our system can be easily integrated into the Alexa cloud service which can take advantage of the fact that the Alexa cloud service has easy access to all the metadata of skills. Figure 5 shows a workflow of our system combined with the Alexa cloud service.

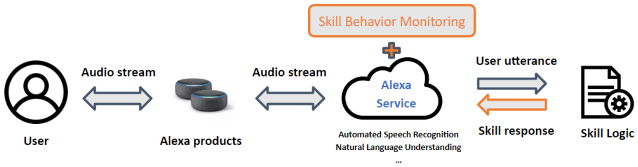


Figure 5: Alexa workflow with our proposed run-time monitoring integrated into cloud service which will monitor every skill response to detect suspicious personal information requests.

4.3 Skill Behavior Check

For each live skill we collected from the Alexa Skill Store, we managed to get a judgment of whether the skill is suspicious or not by performing three steps, including *Question Extraction*, *Permission Check*, and *Consistency Check*. Algorithm 1 shows the procedure of the offline analysis process. *Question Extraction* is line 2. *Permission Check* is from line 3 to line 6. *Consistency Check* is from line 8 to line 18.

4.3.1 ***Question Extraction.*** Skills give all kinds of responses to users to complete interactions for a variety of purposes, such as daily life services (weather forecast, news, map) and entertainment (music, game). We focus on the interactions that involve private information. Our goal is to find out the skill responses asking for personal information because it may introduce privacy risks to users. For example, *"Do you want to continue?"* is not the question we are interested in. We are only going to extract the question like *"What's your zip code?"*.

When a skill wants to ask for some personal information from users, it states what kind of personal information. In other words, the question must contain keywords about that personal information. Thus, we maintain a list of keywords related to the user's personal information as well as their synonyms. For each sentence from the skill's conversation data, our system first conducts a keyword match to get all of the sentences that involve personal information, then determines whether the sentence is interrogative or not. If it is, the system will extract this question for further analysis.

Algorithm 1 Skill Behavior Check

```
1:  for Every response from skill do
2:      if The response is asking for user information then
3:          if Skill claims relevant permission in permission list then
4:              hashValue ← Hash(skillID, permissionList, description, skillResponse)
5:              Add $(hashValue, \text{“}consistent\text{”})$ to database
6:          end if
7:      else
8:          if keyword matched in skill description then
9:              hashValue ← Hash(skillID, permissionList, description, skillResponse)
10:             Add $(hashValue, \text{“}consistent\text{”})$ to database
11:         else
12:             hashValue ← Hash(skillID, permissionList, description, skillResponse)
13:             if Textual entailment module produces ‘entailment’ then
14:                 Add $(hashValue, \text{“}consistent\text{”})$ to database
15:             else
16:                 Add $(hashValue, \text{“}inconsistent\text{”})$ to database
17:             end if
18:         end if
19:     end if
20: end for
```

By manually browsing some skills that ask for the user's information, we found that the skills would ask in basically two ways: WH questions and imperative questions. An example of a WH question is *"What's your name/birthday/city?"*. To identify general WH-questions, we refer to spaCy[14] tags including *"WDT"*, *"WP"*, *"WP$"*, *"WRB"*. Regarding imperative questions, an imperative question usually starts with the verb *"Please tell me your phone number."* To better cover general sentence structures, we built pattern rules as *pattern = ['TAG': 'VB', 'TAG': 'RP', 'OP': '*', 'POS': 'NOUN', 'OP': '*', 'POS': 'ADJ', 'OP': '*', 'POS': 'ADV', 'OP': '*', 'TAG': 'PRP', 'OP': '*', 'TAG': 'PRP$', 'OP': '*', 'POS': 'NOUN']*. Then, when a skill uses a verb to request a noun related to personal information, it will be identified by the question extraction module.

We manually checked the identified skills to make sure they actually asked for personal information. For the skills that did not question personal information, we randomly sampled 100 skills and checked them to ensure we did not miss cases. We iteratively revised the pattern rules to cover edge cases. An example was "What a beautiful name!", which was incorrectly classified as asking for a name. We repeated this review and revision process until we found no edge case.

4.3.2 ***Permission Check.*** Developers are required to build a permission list for their Alexa skills to get the users' personal information serving skill functionalities. Suppose developers have configured the skill this way. In that case, when a user first enables the skill, Alexa asks the user to go to the Alexa app to grant permission to obtain this specific information.

Currently, the available permissions [4] for custom skills related to personal information include Device Address, Customer First Name, Customer Full Name, Customer Email Address, Customer Phone Number, Location Services, and Postal Code.

After our system detects that a skill response is requesting any kind of personal information, it will immediately check if the requested information is in the permission list. If it is, then it means the skill is not malicious because it is following the requirements and asking the users for an explicit grant. In contrast, if the requested information is not in the permission list, the skill becomes suspicious. The detected skill response will be sent to the next system module, i.e., consistency check, in order to determine whether the requested information has any correlation with the skill's functionalities.

4.3.3 ***Consistency Check.*** Even though Amazon's documentation requires the skill to include all of the requested information in the permission list, Amazon's monitoring is enforcing more flexible criteria based on our findings and experiments in Section 3.3. That is, Amazon will determine the skill as legitimate as long as the requested information has a correlation with the skill's functionality. This consistency check module is built to conform to that criteria. A skill asking for personal information, which has a correlation with its functionalities, will be determined as a "consistent skill"; otherwise, it will be an "inconsistent skill."

The skill description given by the developer gives the most abundant information on the skill's functionalities. However, these descriptions have no certain formats and structures. It is a challenge to analyze the skill's description due to the various ways to describe the skill functionalities. A skill's description may involve the keywords of requested personal information. In this case, we apply a keyword-matching approach to check its functionality consistency. However, the description of the benign skill does not necessarily contain the keywords of requested personal information.

To solve the above problem, we apply textual entailment to handle the description texts that cannot be automatically analyzed by keyword-based methods. Textual entailment is used to predict whether, for a pair of texts (text1, text2), the information in the second text can be implied from the first one [16]. The first text is called *premise*, and the second text is called *hypothesis*. If the *hypothesis* can be implied from the *premise*, then the output result is *entailment*. Otherwise, it is *contradiction*. To handle skills' descriptions without information keywords, we consider the skill's description as *premise* and the skill's behavior as *hypothesis*. If the textual entailment model gives an "entailment" result, the skill description implies the skill needs to ask for specific information to serve its functionalities. For example, the skill "Food Hero" has a question asking "To find food near you, I need your permission to view your zip code. What's your zip code?". Its description is "Food Hero will look around your area for highly-rated restaurants and make that annoying decision as to where to eat for you. Let Food Hero pick!" It is easy to understand that this skill needs location information to search nearby restaurants even though the skill provides no permission list. An example that receives a "contradiction" result from textual entailment is the "Ehrlich Pest Control" skill. It asks for a phone number by saying "Please tell me your phone number... area code first." Its description is "The Ehrlich Pest Control skill will tell you the top tips on the prevention of common household pests such as mice, cockroaches, and flies. From tips on cleaning up common feeding sites to the times of day to avoid mosquitoes, these tips will help tackle", which does not imply a phone number is necessary for its functionalities. Textual entailment can understand the semantic information in the descriptions without any keywords being present, which is why we use it to cover the cases that the keyword-matching methods may fail to identify.

Our textual entailment model leverages the AllenNLP [16] research library and a pre-trained ELMo-based Decomposable Attention model [26]. We fine-tuned the model with our skill data. We manually created a labeled dataset of 446 skills, splitting it into 80% training and 20% validation. Three researchers in our group were involved in the labeling process and discussion to agree on the final labels. We applied early stopping [27] for our fine-tuning process, which is a popular technique to avoid overfitting. Our fine-tuned model achieved 99.7% training accuracy and 99.0% validation accuracy. The training loss was 0.008, and the validation loss was 0.01, suggesting that our model converged well and both train/validation performances remained equivalent.

4.4 Run-time Protection

Our run-time protection component monitors the conversation when the user is interacting with the skill and notifies the user if the skill's behavior is inconsistent with the skill's functionalities.

4.4.1 ***Run-time Workflow.*** When the user enables a skill and interacts with it, the system will monitor every skill's interaction with the user. When the skill asks for the user's personal information, the system will analyze if the requested information has a correlation with the skill's functionalities. If the requested information is not related to the functionalities, the system will first send a privacy warning to the user, helping the users be cautious when they interact with the skill. To improve the system's response time, the run-time protection module leverages a database for a quick query. When the monitored skill's behavior matches the record in the database, the system will retrieve the existing result and respond accordingly. When the system encounters a new/modified skill and the skill's behavior has no record in the database, it will perform all the analysis steps and save the result to the database to save time for future checks. For example, suppose user A interacts with a newly published skill *X* which asks for personal information. In that case, the monitoring system will analyze its information request and update the analysis results in the database. Then later, if user B also uses skill *X*, the system can retrieve the matched record from the database very quickly. Algorithm 2 illustrates how the run-time protection component works in detail.

4.4.2 ***Prototype.*** We leveraged Alexa Voice Service (AVS) [6] to implement and test our run-time monitoring approach. The open-source Alexa Voice Service (AVS) enables developers to integrate Alexa directly into any device with a microphone and a speaker, giving the device direct access to cloud-based Alexa capabilities. We instrumented AVS [5] with our skill monitoring design on an Ubuntu machine. In the following section, we discuss the performance overhead of our approach. We also conducted an on-site user study to evaluate the usability of our approach, which is presented in Section 4.6.

4.4.3 ***Overhead.*** We evaluated the performance of the system on the test set of 1,000 skills. In the test set, 200 skills were newly updated which means they had no hash record in the database,

Algorithm 2 Run-time Protection

```
 1: for Every response from skill do
 2:     if The response is asking for user information then
 3:         if Database has record then
 4:             if The record shows it is consistent then
 5:                 Send skill response to the user
 6:             else
 7:                 digest ← (skillID, permissionList, description, skillResponse)
 8:                 hashValue ← Hash(digest)
 9:                 if Textual Entailment module produces ‘entailment’ then
10:                     Add $(hashValue, “consistent”)$ to database
11:                     Send skill response to users
12:                 else
13:                     Add $(hashValue, “inconsistent”)$ to database
14:                     Send privacy alert to the user.
15:                     Send skill response to user.
16:                 end if
17:             end if
18:         else
19:             Go to step 7
20:         end if
21:     else
22:         Send skill response to the user
23:     end if
24: end for
```

among which 50 were asking for personal information through voice without relevant permissions. We used this test set to simulate the real run-time environment of our monitoring system. Overall, the average delay is 0.09s which is negligible. The average delay for keyword matching is 4.8e-06s which can be ignored, while the average delay for textual entailment is 1.38s. Running textual entailment for every interaction might introduce a perceivable delay. To address this challenge, we managed to avoid re-running textual entailment as much as possible with our database approach described in Section 4.4.1.

4.5 Measurement

Employing our proposed system, specifically the Skill Behavior Check component (presented in Section 4.3), we conducted a measurement to identify the suspicious skills published in the Alexa Skills Store.

4.5.1 ***Dataset.*** We collected a dataset of 54,587 skills from the US Alexa Skill Store (as of Jan 2023) based on the methods proposed in previous work [18, 21]. Our dataset includes skill profiles (e.g., description, voice commands, permission list, etc.) and sample conversations with the skills. For each skill, using similar techniques presented in previous studies [18, 21], we collected as many turns of conversation with the skills as possible until receiving repeated content. Regarding the interaction depth for collecting skill responses, we set the minimum to be two turns of conversation, expecting the skills to give more content than just a single welcome message to our invocation. The maximum depth depends on each skill.

4.5.2 ***Results.*** We used our Skill Behavior Check component (presented in Section 4.3) to analyze our dataset. Of 54,587 skills in our dataset, the tool identified 637 skills that requested personal information through voice interaction, among which 142 skills were identified as suspicious (i.e., no permission declaration and no correlation with the functionalities). We manually checked these 142 suspicious skills and found that 141 were correctly detected. The one skill that was false positive was a storyteller skill. This skill told a story in which character A asked character B for his address. Although it was actually a request for personal information, it was not meant to ask the user who was interacting with the skill. However, this might in fact still be an issue if the user does not pay attention or misunderstand. We also manually checked the skills that were not identified as suspicious. We did not find any case that was missed, meaning they either had proper permission declaration in their skill configuration or their behaviors were consistent with their descriptions.

Next, we further examined the identified 141 suspicious skills. Note that we focused on the types of personal information that Amazon considered in their permission list requirement [4], which include: location, email, phone number, and name. We also observed a few skills asking for birthday info. As a result, we found that most skills asked for name (107 skills). 9 skills asked for phone number. 10 skills asked for email address. 8 skills asked for birthday. 7 skills asked for location. In particular, of these 7 location-requesting skills, 4 asked for a specific home address.

4.6 Usability

To evaluate the usability of our defense prototype, we conducted an in-lab user study with 15 participants. Specifically, we studied: (1) the effectiveness of our system, and (2) the efficiency of our system. Our goal for this user study was to evaluate the usability of our proposed run-time skill monitoring system.

4.6.1 ***Methodology.*** We conducted a between-subjects study method in which our participants were split into two groups. The control group (7 participants) interacted with the original AVS system (i.e., the standard Alexa). The experimental group (8 participants) interacted with the instrumented AVS system (i.e., Alexa protected by our proposed system). Participants in the two groups were provided with an identical list of skills to enable and use. The list had three skills (A, B, C) asking for personal information. Skill A was a consistent skill, skill B was an inconsistent skill with a hashed record in the database, and skill C was an inconsistent skill without hashed record in the database. The skills were randomly sampled. After experimenting with the Alexa prototype assigned to them, the participants then answered our questions about the usability including comfort, accuracy, and time delay. Several pilot studies (excluded from our results) were conducted before the actual run to fix errors and ensure data quality.

Our participants were required to be adults who are 18 or older, fluent in English, live in the U.S., and are voice personal assistant users. We were able to reach out to 15 experienced Alexa users as qualified participants via recruiting emails and flyers. Among the 7 participants who tested the original Alexa, 57.1% are male, 28.6% are female, and less than 14.3% are non-binary. Our participants are mostly young adults (57.1% were 18-24 years old) and highly educated (42.9% with a Bachelor's degree and 42.9% with a high school diploma). Among the 8 participants who tested the Alexa with our protection component, 62.5% are male, 25% are female, and less than 12.5% are non-binary. Our participants are mostly young adults (75% were 18-24 years old) and highly educated (75% with a Bachelor's degree). Tables 2 and 3 present the detailed demographic information of our participants in the control group (original Alexa) and the experimental group (protected Alexa), respectively.

Table 2: Original Alexa group (N=7). Demographic information (gender, age, and education) of the participants that used the original Alexa in our user study.

	Participants	Percentage
Gender		
Male	4	57.1%
Female	2	28.6%
Prefer not answer	1	14.3%
Age		
18-24 years old	4	57.1%
25-34 years old	3	42.9%
35 years or older	0	0.0%
Highest level of education completed		
High School Graduate	3	42.9%
Associates Degree	0	0.0%
Bachelor's Degree	3	42.9%
Graduate degree	1	14.3%

Table 3: Protected Alexa group (N=8). Demographic information (gender, age, and education) of the participants who trialed Alexa with our proposed monitoring solution.

	Participants	Percentage
Gender		
Male	5	62.5%
Female	2	25%
Prefer not answer	1	12.5%
Age		
18-24 years old	6	75%
25-34 years old	2	25%
35 years or older	0	0.0%
Highest level of education completed		
High School Graduate	0	0.0%
Associates Degree	0	0.0%
Bachelor's Degree	6	75%
Graduate degree	2	25%

4.6.2 ***Results.*** We evaluate the usability of our proposed system using three metrics: comfort, accuracy, and time delay. In particular, we want to see if the users are comfortable using our system, if the users think our system works accurately, and if the users feel any time delay caused by our system.

85.7% of participants using the original AVS felt uncomfortable when the skill asked for their personal information without mentioning it beforehand. A user added: "The skill should not do some weird stuff like that" when interacting with the inconsistent skills. Most users felt the need for a protection mechanism. In particular, 66.7% of participants thought that Amazon Alexa should improve the protection of user privacy. 75% of the participants who used our proposed system thought that the privacy warnings from our system were helpful.

The system's warnings are accurate based on participants' feedback. The feedback was presented to the participants using a scale from 1 to 5 (strongly disagree to strongly agree). When asked if the system only gives a warning when the skill requests personal information without explicitly disclosing it beforehand, 75% of participants interacting with the protected AVS strongly agreed or agreed with the statement. The remaining 25% held a neutral attitude. No one disagreed. When asked if the system missed warning any suspicious case where the skill asked for unnecessary personal information without mentioning it beforehand, all participants answered "no". Thus, overall our system provides accurate warnings about the suspicious behaviors of the skills.

Our system does not introduce uncomfortable delays to users. The participants were asked to rate the delay using a scale from 1 to 4 (no delay, hard to notice, not obvious, obvious delay). As a result, 62.5% of participants using the protected AVS agreed that the delay was not obvious. The remaining 37.5% thought the delay was hard to notice. The introduced delay is acceptable considering the fact that the original AVS itself naturally gives participants delay feeling to some degree (14% of participants using the original AVS said they felt some delay but not obvious, even though that was without our monitoring system).

5 DISCUSSION

This section discusses the implications, ethical considerations, and limitations of our work. We also propose future research directions, including support for other platforms, non-privacy issues, and users' awareness of the risks.

5.1 Implications and Call for Action

In this paper, we systematically studied Alexa skill vetting and the potential privacy risks. Our findings suggest the following implications and make a call for action to protect VPA consumers from privacy-invasive skills.

Amazon's vetting is inconsistent with their documented policies. It is important to make sure the policies are transparent to developers to avoid unintentional privacy violations in their skills. In addition, the actual vetting process works differently from what is documented, which causes confusion to developers. This might be due to the outdated documentation. However, the actual vetting is more lenient, which allows violating skills to be published. Therefore, VPA service providers need to have stricter enforcement in their vetting. This could be a burden on developers as it might take longer and more complex to get a skill published. However, transparent policies and actionable feedback from the vetting can help to minimize that burden.

Amazon's vetting has some gaps that can be exploited. We showed that Amazon's vetting includes two processes: skill certification and skill monitoring. After a skill passes the skill certification process, it gets published. The adversary can still change the behaviors of the skill by manipulating the backend server. The problem is that the changes will go live instantly. Although the skill monitoring process can help to check such behavior changes, it might not happen as soon as the changes happen. Such gaps are different time windows following a pattern that the adversary can exploit to run malicious skill behaviors without being checked (as also demonstrated in Section 3.3.3 and Section 3.4).

A policy enforcement at run-time is necessary to protect consumers from malicious dynamic behaviors. For skills hosted on the service provider's servers, it is easy to detect behavior changes. Such changes can be held off until passing the skill certification. However, for skills hosted on third-party servers, it is a challenging problem because behavior changes would be unpredictable. Run-time defense approaches such as our proposed system—as a client-side tool or integrated into the VPA system—can effectively detect malicious behavior changes and help users be aware of the risks.

5.2 Ethical Considerations

We did not store or use any user data through the published test skills. We were only interested in what types of requests were made to our test skills instead of what the users said to the skills. We performed an on-site user study to evaluate the proposed skill monitoring system without infringing on participants' privacy. Our work got approval from our Institutional Review Board (IRB). We also contacted Amazon regarding our findings. A representative from the Amazon Alexa Skills Team reached out to us and offered further support for our research.

5.3 Limitations and Future Work

Our proposed skill monitoring system does not design specific methods against adversarial attacks. Adversaries may manipulate skills' descriptions or skills' content to fool the system if they are aware of the underlying techniques of the monitoring system.

In this work, the analysis mainly focuses on the Alexa skills in the US skill store which is the largest market. Future work can further examine skills in other regions, and other virtual personal assistant platforms such as Google Assistant and Apple Siri.

A recent news [3] reports "Amazon's Alexa tells a 10-year-old child to touch penny to exposed plug socket." Previous work also identified inappropriate content and privacy concerns in child-directed Alexa skills [21]. Existing parental control mechanisms were also found insufficient [33]. Hence, the dynamic content of skills may be a threat to children if not properly monitored. Our proposed monitoring approach focuses on detecting suspicious skill behaviors that request personal information. Our consistency check module can be used to inform the necessary permissions and adjust the permission list for each skill accordingly. It can be further extended to cover more kinds of suspicious behaviors such as hate speech, dangerous instructions to kids, etc.

Existing literature showed that smart device users were very concerned and wanted privacy notifications about the data collection activities around them [22, 30]. In our user study, we found that the participants were interested in the privacy warnings given by our proposed system. This indicates that users value their privacy and want to be aware of potential risks from the skills. Thus, future work can focus more on studying users' preferences and designing personalized systems to improve users' awareness.

6 CONCLUSION

We provided a comprehensive understanding of Amazon's skill vetting strategy including its skill certification and monitoring process. Skills can be hosted on third-party servers, which can lead to malicious behaviors. Compared to the prior work, we dug deeper into novel approaches to bypass Alexa skill vetting and perform dynamic behavior changes after the skill gets published. We revealed how adversaries can bypass the vetting and a new attack vector called "versatile intents" with proof-of-concept attacks. We also proposed a run-time skill monitoring system to protect user privacy against such threats.

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation (NSF) Grants (Awards 2323105, 2325369, 2317184) and the Okawa Foundation Research Grant. The US Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright notices thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied of any funding agencies or governments. We thank all participants in our user study for their time and insights. We appreciate the constructive feedback from the anonymous reviewers.

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] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. 2020. Void: A fast and light voice liveness detection system. In 29th {USENIX} Security Symposium ({USENIX} Security 20). 2685–2702.

[3] Amazon. 2022. Amazon's Alexa tells 10-year-old child to touch penny to exposed plug socket. https://www.cnn.com/2021/12/29/business/amazon-alexa-penny-plug-intl-scli/index.html. Accessed: 2022-03-18.

[4] Amazon. 2022. Configure Permissions for Customer Information in Your Skill. https://developer.amazon.com/en-US/docs/alexa/custom-skills/configure-permissions-for-customer-information-in-your-skill.html.

[5] Amazon. 2022. Set Up the AVS Device SDK on Ubuntu. https://developer.amazon.com/en-US/docs/alexa/avs-device-sdk/ubuntu.html.

[6] Amazon. 2022. What is the Alexa Voice Service? https://developer.amazon.com/en-US/docs/alexa/alexa-voice-service/get-started-with-alexa-voice-service.html.

[7] Mary K Bispham, Ioannis Agrafiotis, and Michael Goldsmith. 2019. Nonsense attacks on google assistant and missense attacks on amazon alexa. (2019).

[8] Fabian Braunlein and Luise Frerichs. 2019. Smart Spies: Alexa and Google Home expose users to vishing and eavesdropping. https://www.srlabs.de/bites/smart-spies.

[9] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden voice commands. In 25th {USENIX} Security Symposium ({USENIX} Security 16). 513–530.

[10] Tao Chen, Longfei Shangguan, Zhenjiang Li, and Kyle Jamieson. 2020. Metamorph: Injecting inaudible commands into over-the-air voice controlled systems. In Network and Distributed Systems Security (NDSS) Symposium.

[11] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. 2020. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In 29th {USENIX} Security Symposium ({USENIX} Security 20). 2667–2684.

[12] Long Cheng, Christin Wilson, Song Liao, Jeffrey Young, Daniel Dong, and Hongxin Hu. 2020. Dangerous skills got certified: Measuring the trustworthiness of skill certification in voice personal assistant platforms. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. 1699–1716.

[13] Tamara Denning, Tadayoshi Kohno, and Henry M Levy. 2013. Computer security and the modern home. Commun. ACM 56, 1 (2013), 94–103.

[14] Explosion. 2022. Spacy - industrial-strength natural language processing in python. https://spacy.io/.

[15] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash. 2016. Security analysis of emerging smart home applications. In 2016 IEEE symposium on security and privacy (SP). IEEE, 636–654.

[16] Allen Institute for AI. 2022. Textual Entailment. https://demo.allennlp.org/textual-entailment/elmo-snli.

[17] Nathaniel Fruchter and Ilaria Liccardi. 2018. Consumer attitudes towards privacy and security in home assistants. In Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems. 1–6.

[18] Zhixiu Guo, Zijin Lin, Pan Li, and Kai Chen. 2020. Skillexplorer: Understanding the behavior of skills in large scale. In 29th {USENIX} Security Symposium ({USENIX} Security 20). 2649–2666.

[19] Deepak Kumar, Riccardo Paccagnella, Paul Murley, Eric Hennenfent, Joshua Mason, Adam Bates, and Michael Bailey. 2018. Skill squatting attacks on Amazon Alexa. In 27th {USENIX} Security Symposium ({USENIX} Security 18). 33–47.

[20] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. 2018. Alexa, are you listening? Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. Proceedings of the ACM on Human-Computer Interaction 2, CSCW (2018), 1–31.

[21] Tu Le, Danny Yuxing Huang, Noah Apthorpe, and Yuan Tian. 2022. SkillBot: Identifying Risky Content for Children in Alexa Skills. ACM Trans. Internet Technol. 22, 3, Article 79 (jul 2022), 31 pages. https://doi.org/10.1145/3539609

[22] Tu Le, Alan Wang, Yaxing Yao, Yuanyuan Feng, Arsalan Heydarian, Norman Sadeh, and Yuan Tian. 2023. Exploring Smart Commercial Building Occupants' Perceptions and Notification Preferences of Internet of Things Data Collection in the United States. In 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P). 1030–1046. https://doi.org/10.1109/EuroSP57164.2023.00064

[23] Christopher Lentzsch, Sheel Jayesh Shah, Benjamin Andow, Martin Degeling, Anupam Das, and William Enck. 2021. Hey Alexa, is this Skill Safe?: Taking a Closer Look at the Alexa Skill Ecosystem. In 28th Annual Network and Distributed System Security Symposium (NDSS 2021). The Internet Society.

[24] Song Liao, Christin Wilson, Long Cheng, Hongxin Hu, and Huixing Deng. 2020. Measuring the Effectiveness of Privacy Policies for Voice Assistant Applications. In Annual Computer Security Applications Conference (ACSAC '20). Association for Computing Machinery, New York, NY, USA, 856–869. https://doi.org/10.1145/3427228.3427250

[25] Ajaya Neupane, Nitesh Saxena, Leanne M Hirshfield, and Sarah E Bratt. 2019. The Crux of Voice (In) Security: A Brain Study of Speaker Legitimacy Detection.. In NDSS.

[26] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933 (2016).

[27] Lutz Prechelt. 2012. Early Stopping — But When? Springer Berlin Heidelberg, Berlin, Heidelberg, 53–67. https://doi.org/10.1007/978-3-642-35289-8_5

[28] Grand View Research. 2020. Intelligent Virtual assistant market size report. https://developer.amazon.com/en-US/docs/alexa/custom-skills/configure-permissions-for-customer-information-in-your-skill.html.

[29] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. 2020. Light commands: laser-based audio injection attacks on voice-controllable systems. In 29th {USENIX} Security Symposium ({USENIX} Security 20). 2631–2648.

[30] Parth Kirankumar Thakkar, Shijing He, Shiyu Xu, Danny Yuxing Huang, and Yaxing Yao. 2022. "It Would Probably Turn into a Social Faux-Pas": Users' and Bystanders' Preferences of Privacy Awareness Mechanisms in Smart Homes. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 404, 13 pages. https://doi.org/10.1145/3491102.3502137

[31] Dawei Wang, Kai Chen, and Wei Wang. 2021. Demystifying the Vetting Process of Voice-controlled Skills on Markets. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5, 3 (2021), 1–28.

[32] Qi Wang, Pubali Datta, Wei Yang, Si Liu, Adam Bates, and Carl A Gunter. 2019. Charting the attack surface of trigger-action IoT platforms. In Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. 1439–1453.

[33] Peiyi Yang, Jie Fan, Zice Wei, Haoqian Li, Tu Le, and Yuan Tian. 2023. Towards Usable Parental Control for Voice Assistants. In Proceedings of Cyber-Physical Systems and Internet of Things Week 2023 (San Antonio, TX, USA) (CPS-IoT Week '23). Association for Computing Machinery, New York, NY, USA, 43–48. https://doi.org/10.1145/3576914.3587491

[34] Jeffrey Young, Song Liao, Long Cheng, Hongxin Hu, and Huixing Deng. 2022. SkillDetective: Automated Policy-Violation Detection of Voice Assistant Applications in the Wild. In 31st USENIX Security Symposium (USENIX Security 22). USENIX Association, Boston, MA, 1113–1130. https://www.usenix.org/conference/usenixsecurity22/presentation/young

[35] Xuejing Yuan, Yuxuan Chen, Aohui Wang, Kai Chen, Shengzhi Zhang, Heqing Huang, and Ian M Molloy. 2018. All your alexa are belong to us: A remote voice control attack against echo. In 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 1–6.

[36] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. 2019. Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 1381–1396.

[37] Yangyong Zhang, Lei Xu, Abner Mendoza, Guangliang Yang, Phakpoom Chinprutthiwong, and Guofei Gu. 2019. Life after speech recognition: Fuzzing semantic misinterpretation for voice assistant applications. In Proc. of the Network and Distributed System Security Symposium (NDSS'19).