Analyzing Privacy Vulnerabilities of Third-Party Skills in Alexa

Shivaprakash Balasubramanian
sbalas22@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Varsha Anantha Ramu Sharma
vananth4@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

1 INTRODUCTION

Smart speakers such as Amazon Echo and Google Home have gained immense popularity around the world, mostly due to the benefits brought by the built-in virtual personal assistants (VPA). These VPA services offer interactive actions through voice commands provided by the user. In addition to the built-in capabilities, these VPA services can be further extended by using third-party skills. Similar to Android and iOS applications that are available on Google Play Store and App Store, third-party skills are made available on Google and Amazon marketplace, attracting users as well as developers with malicious intent. Researches in the recent past have discovered that malicious developers have been able to route users' requests to malicious skills without the users' consent. This has been achieved by creating skills that have similar names as that of the legitimate ones. The scope of this project is to implement a system that systematically explores the interactive behavior of skills developed for Alexa (the built-in VPA present on Amazon Echo) and identify the ones which violate the users' privacy. The system categorizes a skill as malicious if it collects users' private information without following the standard developer specifications as demanded by the Amazon marketplace.

Code: alexa-skill-privacy

2 PROBLEM MOTIVATION

Alexa is one of the leading virtual personal assistants (VPA) with over 100 million active devices around the world. Although VPAs have gained great popularity and are widely used across households, their privacy violations pose a serious threat, particularly to Amazon's Alexa. The large user base and the importance of protecting the user's right to privacy is prominent, which motivates us to build a system that is impactful and relevant to almost everyone. One of Alexa's promising avenues is the ability for developers to build third-party skills and form something similar to that of an open-source community. To place these interactive devices that listen to users 24x7 requires a lot of trust gaining from Alexa's perspective and it is incumbent to ensure the privacy of the customer is not violated. To the best of our knowledge, there is no prior research apart from the one mentioned in the reference paper that systematically explores the interactive behavior of these third-party skills. This is mainly due to the challenges that arise while handling the skills' input/output which is mostly in the form of natural languages. Given the large number of users who interact with Alexa through these skills, we believe there has to be better transparency when it comes to how much of the users' private information the skill owners can access. Our motivation to build a skill explorer system arises from these problems.

3 RELATED WORK

Recent studies to explore the privacy issues in Virtual Private Assistants (VPAs) have focused on understanding the attacks on skills and the attacks on smart speakers. Studies have been carried out to understand the invocation mechanisms of skills. KUMAR et al. discover skill squatting, a kind of homo-phonic attack to divert users' requests to an undesired skill [8]. Furthermore, Zhang et al. find voice squatting and voice masquerading, which routing the users away from legitimate skills to similar-sounding skills [9]. The main focus of these studies is the invocation mechanism of skills, while the reference paper implemented in this project explores the behaviors of Alexa skills and performs analysis on the content of the conversation to categorize malicious skills. Additionally, some studies have explored the attacks on smart speakers. Studies [10, 11, 12] have analyzed the security and privacy of IoT devices in general which include smart speakers as well. Carlini et al. [13] performed hidden voice attacks on Amazon Echo, which confirms the feasibility of performing audio attacks on the device. However, these studies, including Bispham et al. [14] demonstrate how to inject commands into smart speakers or related audio speech recognition (ASR) systems without being captured by human users, while the reference paper is implemented in this project studies the interactive behaviors of these skills. Although the concept of invocation of skills has been recently explored to identify malicious skills, less is understood about the contents provided by skills or their behaviors. In the reference paper, to the best of our knowledge, there has not been any prior research to systematically explore the behaviors of skills. This is mostly because an Alexa skill is similar to a web service, fully black-box to the analyzer and the inputs/outputs to the skill are in the form of natural languages. The contributions of our project is to employ the methodologies specified in the reference paper to build a system that systematically studies the interactive behavior of Alexa skills using grammar-based techniques and flag malicious skills which violate the users' privacy.

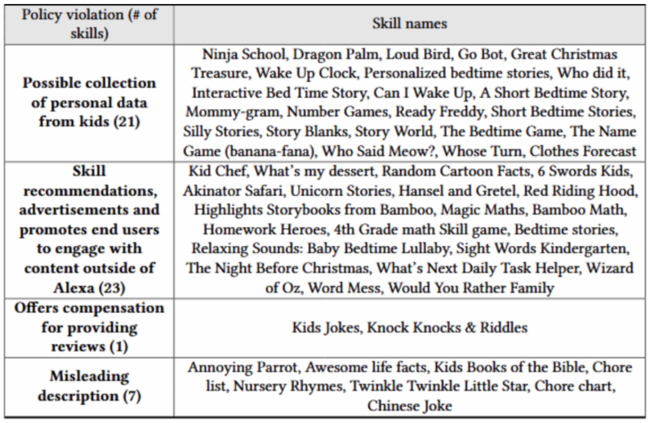
4 PROPOSED APPROACH

4.1 Data Collection

According to the study in the reference paper, there has not been any prior research carried out to systematically explore the interactive behavior of skills developed for Alexa. This is due to the following challenges:

(1) Fully Black-box: A skill is similar to that of a web service, which is fully black box to the analyzer. The analyzer can only provide inputs to the skill and observe the responses. The internal processing and the various states of the model are unknown since the analyzer has no access to the service.
(2) Skill interaction is in the form of natural languages: To explore the interactive behavior of a skill, the analyzer needs to understand the questions asked by Alexa and sort out certain answers in natural languages.

To better understand skill interactions, we manually explored the different skills listed on the Amazon marketplace. As the marketplace has over 100K skills, to identify the skills which violate the users' privacy, we referred to paper [4] which was presented as part of FTC's PrivacyCon 2020 virtual conference. In this paper, the authors have identified the skills that violate users' privacy and classified them as "possible personal data collection", "skill recommendation and advertisements outside the content description", "offers compensation for providing reviews" and "misleading description". However, we could not locate most of the skills listed in [Figure 1] on the marketplace, which implies they were taken down in the recent past by Amazon. Furthermore, we read through the paper [5] to look for prospective skills that violate the developer specifications as demanded by Amazon. Therefore, manually locating the skills which invade users' privacy was one of the key challenges we faced during the process of data collection.

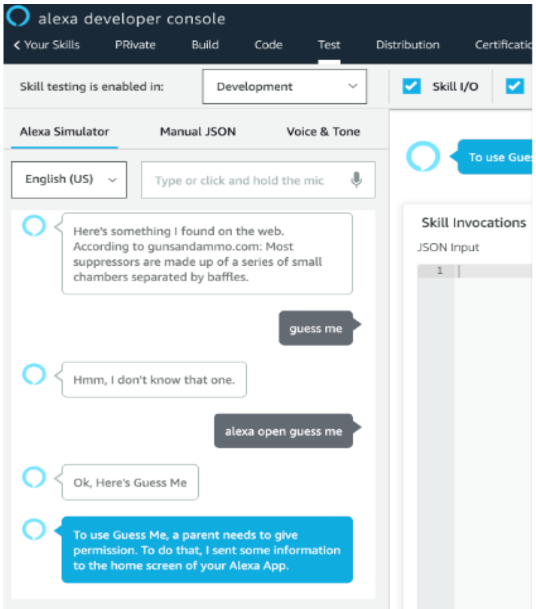


Policy violation (# of skills)	Skill names
Possible collection of personal data from kids (21)	Ninja School, Dragon Palm, Loud Bird, Go Bot, Great Christmas Treasure, Wake Up Clock, Personalized bedtime stories, Who did it, Interactive Bed Time Story, Can I Wake Up, A Short Bedtime Story, Mommy-gram, Number Games, Ready Freddy, Short Bedtime Stories, Silly Stories, Story Blanks, Story World, The Bedtime Game, The Name Game (banana-fana), Who Said Meow?, Whose Turn, Clothes Forecast
Skill recommendations, advertisements and promotes end users to engage with content outside of Alexa (23)	Kid Chef, What's my dessert, Random Cartoon Facts, 6 Swords Kids, Akinator Safari, Unicorn Stories, Hansel and Gretel, Red Riding Hood, Highlights Storybooks from Bamboo, Magic Maths, Bamboo Math, Homework Heroes, 4th Grade math Skill game, Bedtime stories, Relaxing Sounds: Baby Bedtime Lullaby, Sight Words Kindergarten, The Night Before Christmas, What's Next Daily Task Helper, Wizard of Oz, Word Mess, Would You Rather Family
Offers compensation for providing reviews (1)	Kids Jokes, Knock Knocks & Riddles
Misleading description (7)	Annoying Parrot, Awesome life facts, Kids Books of the Bible, Chore list, Nursery Rhymes, Twinkle Twinkle Little Star, Chore chart, Chinese Joke

Figure 1: List of skills with privacy violation in Alexa's skill store as mentioned in the paper[4]

Along with using pre-existing research to shortlist the skills from the Amazon marketplace, we considered some additional parameters such as the popularity of the skill, the developer history, and the outcome of soft testing the skills on Amazon Echo devices. The user ratings and the number of downloads were considered while shortlisting skills based on popularity. To ensure the shortlisted skills form a representative subset, we selected skills that have high (4.5 stars and above), medium (2.5 to 4 stars), and low (2 stars and below) ratings. To determine whether a correlation exists between the developer history and privacy invasion, we tried to shortlist skills published by developers whose skills have been previously flagged as malicious. Lastly, a portion of them were randomly selected based on the soft testing performed using the Amazon Echo device.

We discovered that although skills are similar to mobile applications, they have certain key differences. One such difference is the way to request for services, i.e, to interact with skills users need to provide voice commands whereas for mobile applications it's mostly click operations. Therefore, to build an interactive system, one possible approach is to feed the utterances and the answers directly to Amazon Echo, then record the outputs and transform them into texts by using speech recognition tools such as Google TTS. As this approach is cumbersome, we decided to use the simulator on Alexa Developer Console, which is often used by developers to test their skills. The 'Test' section in the console consists of a simulator that allows developers to interact and communicate with skills using texts [Figure 2].



Here's something I found on the web. According to gunsandammo.com: Most suppressors are made up of a series of small chambers separated by baffles.

guess me

Hmm, I don't know that one.

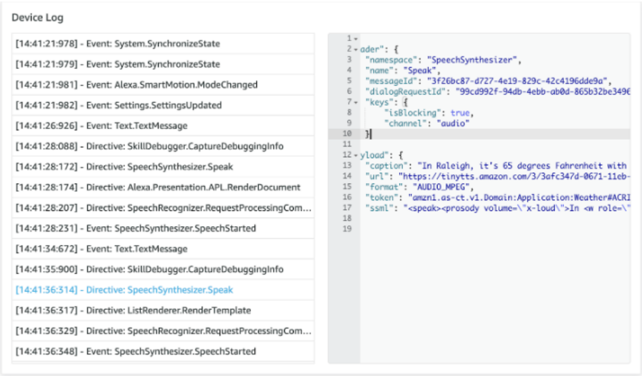
alexa open guess me

Ok, Here's Guess Me

To use Guess Me, a parent needs to give permission. To do that, I sent some information to the home screen of your Alexa App.

Figure 2: Skill interaction on the Alexa Developer console using the simulator

As the interactions are stored as part of the device logs [Figure 3] on the developer console, web scraping was used to obtain them in a plain text format. Additionally, our system leverages PolicyLint to convert the HTML page containing the privacy policy to plaintext. The pre-processing carried out by PolicyLint allows us to perform deeper natural language processing such as POS tagging, dependency parsing, etc.

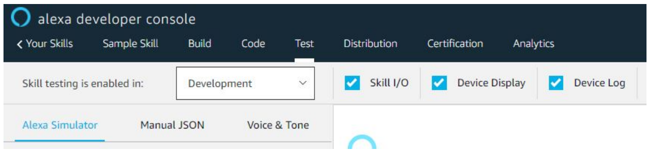


```
[14:41:21:978] - Event: System.SynchronizeState
[14:41:21:979] - Event: System.SynchronizeState
[14:41:21:981] - Event: Alexa.SmartMotion.ModeChanged
[14:41:21:982] - Event: Settings.SettingsUpdated
[14:41:26:926] - Event: Text.TextMessage
[14:41:28:088] - Directive: SkillDebugger.CaptureDebuggingInfo
[14:41:28:172] - Directive: SpeechSynthesizer.Speak
[14:41:28:174] - Directive: Alexa.Presentation.APL.RenderDocument
[14:41:28:207] - Directive: SpeechRecognizer.RequestProcessingCom...
[14:41:28:231] - Event: SpeechSynthesizer.SpeechStarted
[14:41:34:672] - Event: Text.TextMessage
[14:41:35:900] - Directive: SkillDebugger.CaptureDebuggingInfo
[14:41:36:314] - Directive: SpeechSynthesizer.Speak
[14:41:36:317] - Directive: ListRenderer.RenderTemplate
[14:41:36:329] - Directive: SpeechRecognizer.RequestProcessingCom...
[14:41:36:348] - Event: SpeechSynthesizer.SpeechStarted
```

```

ader": {
  "namespace": "SpeechSynthesizer",
  "name": "Speak",
  "messageId": "3f26bc87-d727-4e19-829c-42c4196dde9a",
  "dialogRequestId": "99cd992f-94db-4ebb-ab0d-865b32be3496
  "keys": {
      "isBlocking": true,
      "channel": "audio"
  }

ylоad": {
  "caption": "In Raleigh, it's 65 degrees Fahrenheit with
  "url": "https://tinytts.amazon.com/3/3afc347d-0671-11eb-
  "format": "AUDIO_MPEG",
  "token": "amzn1.as-ct.v1.Domain:Application:Weather#ACRI
  "ssml": "<speak><prosody volume=\"x-loud\">In <w role=\"


```

Figure 3: Device logs of skill interaction from the Alexa Developer Console

To interact with a skill using the simulator, login to the Alexa Developer Console using the Amazon account credentials, and navigate to the "Alexa Skills Kit" section. As third-party skills can not be searched and tested directly on the simulator using neither the skill name nor the skill identifier, it is essential to create a sample skill of your own. This can in turn be used to invoke other third-party skills. In the "Skills" tab, click on the "Create skill" option to initiate the creation of a new skill. Provide an appropriate skill name and select the language of interaction you would prefer. Select the default options for the rest of the steps and complete the skill creation process. Once the skill has been successfully created, navigate to the "Test" tab, and ensure the options selected to resemble the example shown in [Figure 4].



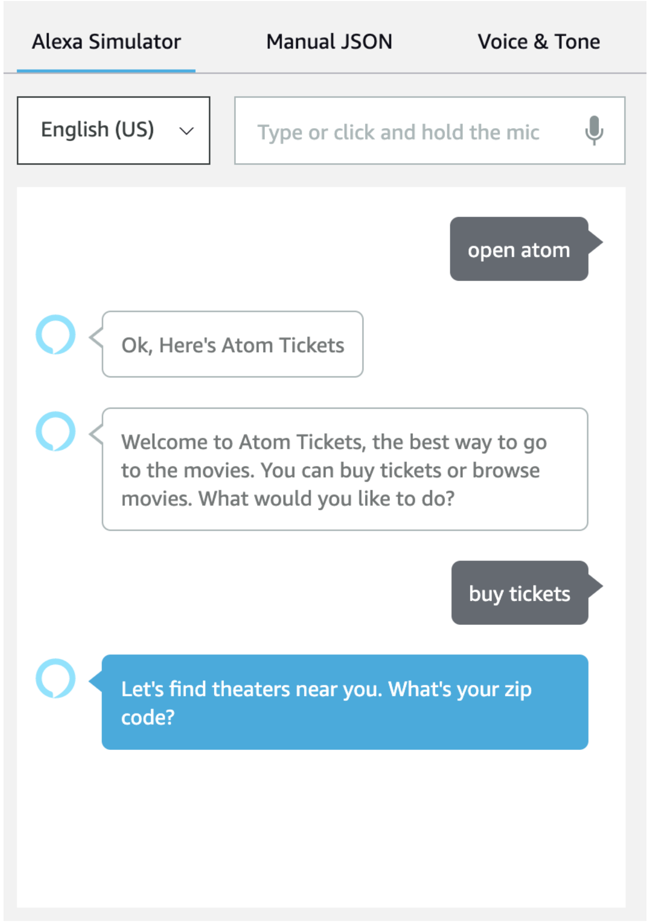
Skill testing is enabled in: Development
☑ Skill I/O ☑ Device Display ☑ Device Log

Figure 4: Simulator options to be enabled on Alexa Developer Console

To invoke a third-party skill using the simulator, you can take advantage of the sample interactions and utterances specified as part of the skill description in the

Amazon marketplace. For instance, the utterance mentioned in the description of a skill named "Age Smart" is "Alexa, open age smart". To invoke this skill, you can type "open age smart" in the input box and begin interacting with the skill as shown in [Figure 5]. Furthermore, to enhance the process of data collection, we tried to cover all the possible branches of the interaction tree using the simulator. This provides an opportunity to evaluate the ground truth and validate the results of the system under development. This interaction on the console can be extracted as device logs from the console. The speechlet directive [SpeechSyntesizer.Speak] shown in the [Figure 6] is the log of the voice that Alexa interacts with the user. On interacting with multiple skills, we determined that some of them have to be manually enabled using the Alexa app beforehand. To enable a skill, download the "Amazon Alexa" app on the mobile phone, browse for a skill by its name and click on "Enable to Use". The logs are further run against the NLP parser to check if the skill requests any personal information.

Furthermore, as we need to verify whether a skill abides by the developer specification, the skill description is scraped from the Amazon marketplace. The system leverages an external tool named PolicyLint to convert the HTML page containing the privacy policy to plaintext. The pre-processing carried out by this tool allows us to perform deeper natural language processing such as POS tagging, dependency parsing, etc on the skill description. We repeat the process for the 100 skills selected and perform a sample set analysis of our results with the ground truth we have created.



open atom

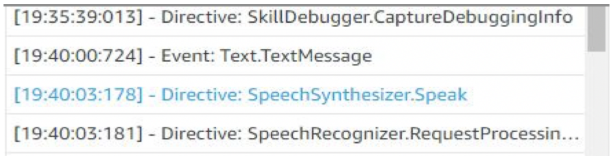
Ok, Here's Atom Tickets

Welcome to Atom Tickets, the best way to go to the movies. You can buy tickets or browse movies. What would you like to do?

buy tickets

Let's find theaters near you. What's your zip code?

Figure 5: Sample skill utterance used for initiating interaction



```
[19:35:39:013] - Directive: SkillDebugger.CaptureDebuggingInfo
[19:40:00:724] - Event: Text.TextMessage
[19:40:03:178] - Directive: SpeechSynthesizer.Speak
[19:40:03:181] - Directive: SpeechRecognizer.RequestProcessin...
```

Figure 6: Speech Directives in Alexa Developer Console

4.2 System Design

According to the rules of Amazon marketplace, developers are allowed to collect some forms of personal information to enhance the user experience. Such information includes the user's name, email address, phone number, etc. and this should be claimed in the privacy policy of skills [15]. In the privacy policy, the developers are required to clearly state what kind of personal information is being collected. However, in the reference paper the authors claim that some developers collect such information but do not mention it in the privacy policy. The interactive skill content obtained from the Alexa Developer console can be analyzed to detect such illegal collection of information and flag the malicious third-party skills. [Figure 7] represents the proposed system design carry out this task.

First to interact with a skill developed for Alexa, we decided to use the developer console as it is easier to obtain the interactive content in the form of device logs. All possible branches of the conversation were covered for each skill to get the complete information. This helps avoid text to speech (TTS) conversion which

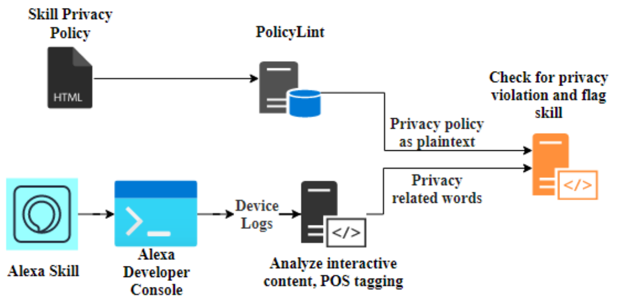
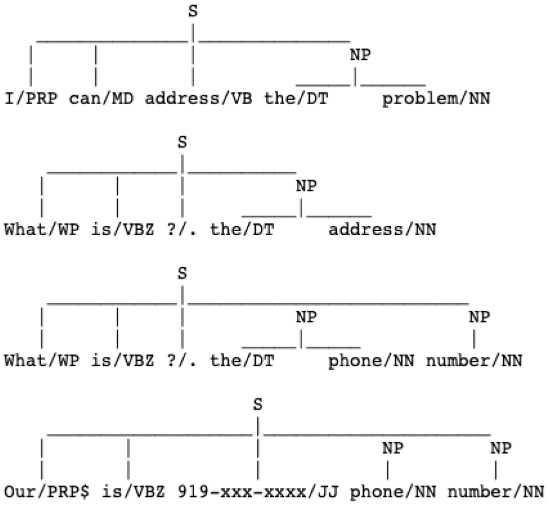

Skill Privacy Policy
HTML
PolicyLint
Check for privacy violation and flag skill
Privacy policy as plaintext
Privacy related words
</>
Alexa Skill
Alexa Developer Console
Device Logs
</>
Analyze interactive content, POS tagging

Figure 7: System design to flag a privacy invading skill if it violates the developer specifications

is cumbersome and likely more error prone. Web scraping was used to obtain the interactive content in the plaintext format. This is provided as an input to the "nlp_parser.py".

We cannot directly compare the privacy words in the interactive content and flag them as negative because this would result in false negatives. For example, if a skill lists the phone number of restaurants around you, "The restaurant number is xxx-xxx-xxxx". Identifying just the keyword "phone number" would result in an incorrect flagging of the skill. To distinguish the two situations, we leverage the dependency based parse tree [fig 6] where all the nodes are words. In the "nlp_parser.py" file, we check the part of speech of the word, for eg, if "address" is used as a verb, the skill wouldn't be marked for flagging. Also, the owner of the verb is taken into consideration before flagging the skill. After extracting the conversation from skill interaction, we use the POS tags to identify the context of the privacy related word used in the sentence. The privacy related words which appear in the right context are further provided as an input to "skill_checker.py".

On obtaining the privacy keywords that a skill requests from users, we further examine whether the skill conflicts with the developer specifications. We first check if the privacy related keywords are mentioned in the privacy policy. If no such declaration is found, the skill is viewed as conflicting with developer specification. However, in real situations, it is hard to check whether the privacy keywords are used for requesting users' data. For instance, "We collect users' personal information including their name and email", where the general term "personal information" and the pronoun



```
                    S
    ________________|_________________
   |        |       |                 NP
   |        |       |             ____|______
I/PRP can/MD address/VB the/DT     problem/NN

                  S
    ______________|_______________
   |      |       |               NP
   |      |       |           ____|_____
What/WP is/VBZ ?/. the/DT     address/NN

                  S
    ______________|_____________________________
   |      |       |               NP            NP
   |      |       |           ____|_____        |
What/WP is/VBZ ?/. the/DT     phone/NN number/NN

                          S
    ______________________|________________________
   |            |         |                NP        NP
   |            |         |                |         |
Our/PRP$ is/VBZ 919-xxx-xxxx/JJ phone/NN number/NN
```

Figure 8: Parse tree of the extracted sentences from different skills

make the analysis challenging. Specifically, what makes the situation complex is the general declaration which usually contains three types of words to collect users' information, including a verb of collect information (e.g., collect, gather, check), a general term (e.g., personal information, personal data), and subsumptive relationships words (e.g., such as, include). As the privacy policies are HTML pages, we leverage PolicyLint[7] to convert the HTML page to plaintext. The pre-processing carried out by PolicyLint allows us to perform deeper natural language processing such as POS tagging, dependency parsing etc. Therefore, in our implementation, if any of the two of three types of words are in a declaration of "ngram", the skill is not flagged as a malicious skill. Based on this, the "skill_checker.py" declares whether a skill violates the users' privacy or not.

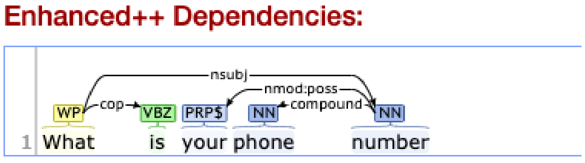


Figure 9: Dependency tree of the extracted sentences using Standford CoreNLP

Another algorithm we use is to verify if the skill is listening to conversations even after the user has ended the interaction with a particular skill. This is being tested out manually on Alexa devices. Skills usually have to stop interacting/listening to the user after the “STOP” word is uttered. This is one of the basic regulations all skills have to follow. Some of the skills break this rule and continue to listen to user conversations and have other keywords to trigger the stop functionality. Then we leverage some built-in functions of the virtual personal assistant (VPA) and check whether the VPA is activated. For example, we ask the time using Alexa’s own function “what time”. If the response is the current time, we can verify that the skill has already exited. We have tested it on the above mentioned 5 skills but all of them abide by the policies and regulations.

5 EVALUATION AND RESULTS

While validating the approach of flagging the skills we begin with measuring it’s accuracy. After analyzing 100 skills, the system flagged 16 skills as malicious. We manually checked the results of the comparison between the privacy related keywords and the privacy policies, and found that there were 82 true positives and 18 false positives. One of the statements used in a false positive flagged privacy policy was "We may store information such as names, email addresses, telephone number, profile picture, user survey responses, third party social media account information". As the number of general declaration types in the statement is 0 according to the declaration types considered for the system, it flagged the policy. Since the privacy policies can be very diverse when framed by different developers, it might be challenging to categorize all the declaration types, which further causes these false positives.

The NLP parser which uses dependency trees from Stanford coreNLP and POS tags has an accuracy of 100% when tested against a small subset of 7 skills. However, when tested against a larger and diverse skill set, it might give rise to a few false positives. In the 4 examples, as shown in [Figure 8], it correctly flags the skill which uses the sentence "What is your phone number?" as something that violates the policy documents because number is a privacy-related word and the object whose number is being requested is the Alexa end user. Additionally, the system functions as intended by not flagging a sentence "Our phone number is xxx", where the skill is providing some information to the end user instead of collecting any personal information related to the user.

The second algorithm involves directly interacting with the skill on the physical device and asking Alexa for the time after we utter the “STOP” phrase as mentioned in the skill description. Some skills breach this by continuing to listen to the conversation and give the wrong time when the user requests for current time (time of skill invocation). This is a direct metric used to identify if the skill continues to listen after it is supposed to terminate and does not require any other validation or evaluation. We tested this during every skill interaction, but all of them abide by the regulations, i.e, they responded with the correct time, suggesting that they actually terminate when the "STOP" phrase is uttered.

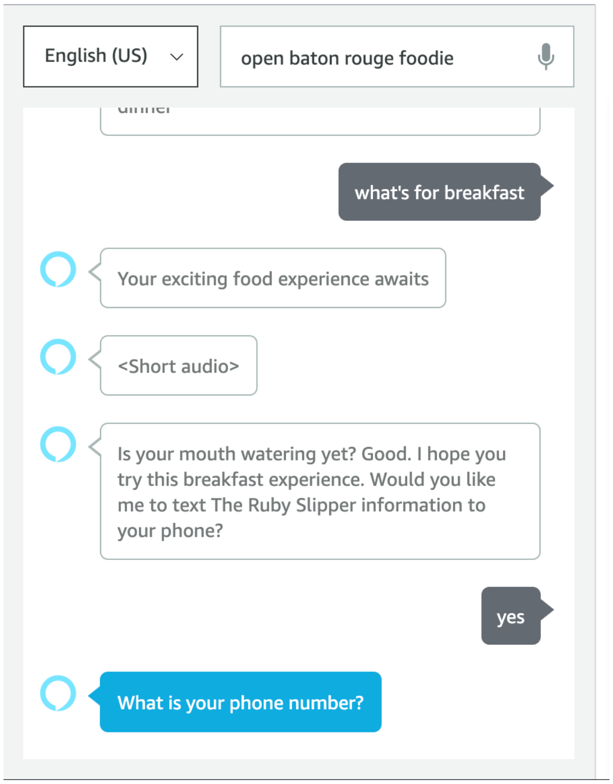
We evaluate our algorithm with the help of the ground truth information collected for the skills during the manual interaction phase through the Alexa developer console.

	Actual Good Skills	Actual Malicious Skills
Good Skills	68	16
Skills Flagged as Malicious	2	14

Table 1: Results of testing the system against 100 skills

The results of testing out the system against 100 skills from the Amazon skill store are presented in [Table 1]. We begin by validating the accuracy of our approach employed to flag privacy-violating skills. On analyzing 100 skills, our system found 16 skills conflicting with the developer specifications. Out of these 16 skills, 8 skills do not mention that they are collecting any form of personal information in their privacy policy or do not have valid privacy policies, 6 skills conflict with the claimed privacy policy and 2 skills are false positives which do not conflict with their privacy policies. Our system found 84 skills that abide by the developer specification. However, 16 of these turned out to be false positives.

A total of 30 false positives are mainly due to the variations in the declarative types. For instance, after converting one of the privacy policies from HTML to plain text using PolicyLint, the phrase obtained is "..you may be asked to enter your name, email address or other details to help you with your experience", as our system only contains words such as "collect, gather, check" in the declaration type "collect_info", it flags the skill as malicious incorrectly although the privacy policy does not have conflicting specifications. The skill named "Baton Rouge Foodie" shown in [Figure 9] is one such false positive example, where the skill requests for the user's phone number and mentions it in their privacy policy but is flagged as a malicious skill. Therefore, the inclusiveness of the declaration types can be further enhanced for better results and provide scope for future work on the system. Additionally, the most frequently requested information are name, phone number and address.



open baton rouge foodie

dinner

what's for breakfast

Your exciting food experience awaits

<Short audio>

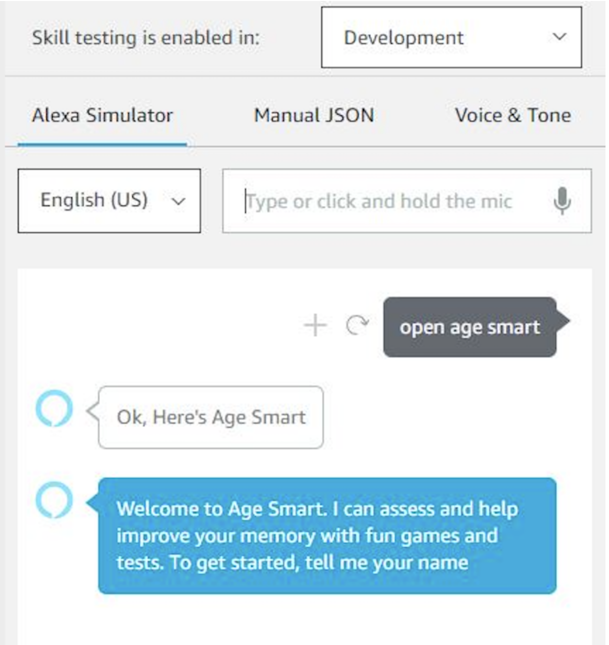
Is your mouth watering yet? Good. I hope you try this breakfast experience. Would you like me to text The Ruby Slipper information to your phone?

yes

What is your phone number?

Figure 10: Example of a skill which is incorrectly flagged as malicious

During our analysis, we came across multiple skills which collect user's information but do not have an authentic privacy policy listed on the Amazon market place. For instance, the skill "Age Smart" shown in the [Figure 10], requests for user's name but redirects to a Privacy Policy that is non-existent.

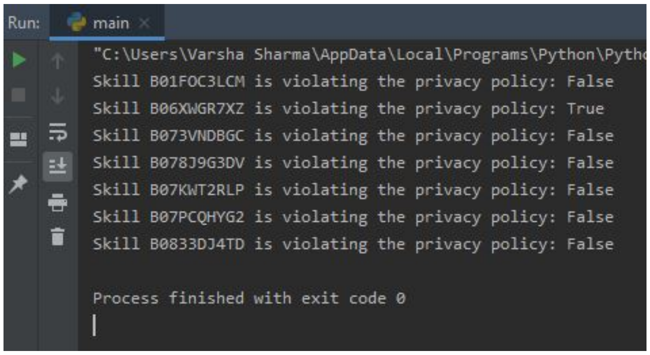

Skill testing is enabled in: Development

open age smart

Ok, Here's Age Smart

Welcome to Age Smart. I can assess and help improve your memory with fun games and tests. To get started, tell me your name

Figure 11: Example of a skill which requests for personal information but does not have a legitimate privacy policy



```
"C:\Users\Varsha Sharma\AppData\Local\Programs\Python\Pytho
Skill B01FOC3LCM is violating the privacy policy: False
Skill B06XWGR7XZ is violating the privacy policy: True
Skill B073VNDBGC is violating the privacy policy: False
Skill B078J9G3DV is violating the privacy policy: False
Skill B07KWT2RLP is violating the privacy policy: False
Skill B07PCQHYG2 is violating the privacy policy: False
Skill B0833DJ4TD is violating the privacy policy: False

Process finished with exit code 0
```

Figure 12: Output of the system when tested against a subset of 7 skills

6 LIMITATIONS

As mentioned in Section 4.1, one of the key challenges we faced during data collection was manually locating the skills which violate developer specifications by not declaring the information collection in their privacy policy. This makes it quite challenging to completely evaluate the system. Another challenge was Amazon's flagging policy was being rigid, it was difficult to find skills that actually violated user privacy as the ratios are very marginal. Getting the ground truth for the skills has no shortcut, but to manually go through the complete parse tree of the interactions of the skill and evaluate.

7 FUTURE WORK

Currently, the system flags malicious skills which do not abide by the developer specifications while collecting users' personal information, we believe another potential check to flag skills would be to see if the information requested by the skill is in accordance to it's description provided by the developers. Furthermore, we believe that user ratings and developer history play an important role in identifying a malicious skill. Therefore, the future work can focus on considering these two factors in addition to developer specifications to flag skills which invade users' privacy.

In the current implementation of the paper, the NLP parser and the PolicyLint code work independent of each other and we manually use the results of the same to evaluate the skill. This process can be automated by identifying the persona information in both the outputs and matching them to directly flag the skill. Also with information about the cohesive meaning of the sentence it is easier to interpret the meaning of the PolicyLint results.

8 INDIVIDUAL CONTRIBUTIONS

The project work progress and the individual contributions are as follows:

Task	Owner
Reference paper reading and literature survey	Shiva, Varsha
Interacting with third-party Alexa Skills to explore if any privacy related information is being collected	Shiva
Reach to the authors, understand the working of Alexa developer console and certain nuances in the paper	Shiva
Perform keyword-based search on the TTS responses to detect possible privacy invasions based on the privacy policy	Varsha
NLP POS tagging to flag skills conflicting with the developer specifications in the skill description	Shiva, Varsha
Extract skill interaction device logs from Alexa Developer Console	Shiva
Set up a web scraper to extract policy information of skills on the store using PolicyLint	Varsha
Explore the stop skill feature on physical devices	Varsha

REFERENCES:

1. SkillExplorer: Understanding the Behavior of Skills in Large Scale
2. https://www.usenix.org/conference/usenixsecurity20/presentation/guo
3. https://developer.amazon.com/en-US/docs/alexa/alexa-skills-kit-sdk-for-nodejs/develop-your-first-skill.html
4. https://www.zdnet.com/article/academics-smuggle-234-policy-violating-skills-on-the-alexa-skills-store/
5. www.usenix.org/system/files/conference/usenixsecurity18/sec18-kumar.pdf
6. https://github.com/varsha5595/csc533-project
7. Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. Policylint: Investigating internal privacy policy contradictions on google play. In Nadia Heninger and Patrick Traynor, editors, 28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019, pages 585–602. USENIX Association, 2019.
8. Deepak Kumar, Riccardo Paccagnella, Paul Murley, Eric Hennenfent, Joshua Mason, Adam Bates, and Michael Bailey. Skill squatting attacks on amazon alexa. In William Enck and Adrienne Porter Felt, editors, 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018, pages 33–47. USENIX Association, 2018.
9. Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFengWang, 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 2019, San Francisco, CA, USA, May 19-23, 2019, pages 1381–1396. IEEE, 2019.
10. Tamara Denning, Tadayoshi Kohno, and Henry M. Levy. Computer security and the modern home. Commun. ACM, 56(1):94–103, 2013.
11. Earlence Fernandes, Jaeyeon Jung, and Atul Prakash. Security analysis of emerging smart home applications. In IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016, pages 636–654. IEEE Computer Society, 2016.
12. Nathaniel Fruchter and Ilaria Liccardi. Consumer attitudes towards privacy and security in home assistants. In Regan L. Mandryk, Mark Hancock, Mark Perry, and Anna L. Cox, editors, Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018. ACM, 2018.
13. Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David A. Wagner, and Wenchao Zhou. Hidden voice commands. In Thorsten Holz and Stefan Savage, editors, 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016, pages 513–530. USENIX Association, 2016.
14. Mary K. Bispham, Ioannis Agrafiotis, and Michael Goldsmith. Nonsense attacks on google assistant and

missense attacks on amazon alexa. In Paolo Mori, Steven Furnell, and Olivier Camp, editors, Proceedings of the 5th International Conference on Information Systems Security and Privacy, ICISSP 2019, Prague, Czech Republic, February 23-25, 2019, pages 75–87. SciTePress, 2019.

15. https://developer.amazon.com/zh/docs/custom-skills/request-customer-contact-information-for-use-in-your-skill.html.