AUTOMATIC TESTING OF SMART SPEAKER APPS

J.L.A.I.A.Sandaruwani
198772M

Faculty of Information Technology
University of Moratuwa

July 2022

AUTOMATIC TESTING OF SMART SPEAKER APPS

J.L.A.I.A.Sandaruwani
198772M

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfillment of the requirements of Degree of Master of Science in Information Technology

July 2022

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student — Signature of Student

UOM Verified Signature

J.L.A.I.A.Sandaruwani …….

Date:

Supervised by

Name of Supervisor — Signature of Supervisor

Dr. Kulani Mahadewa ……

Date:

Acknowledgment

This research behind it would not have been possible without the exceptional support of my supervisor, Dr. Kulani Mahadewa. Her enthusiasm, knowledge, and exacting attention to detail have been an inspiration and kept my work on track.

And also, thank you to Ms. Rrubaa Panchendrarajan for her guidance, supervision, advice, sparing valuable time, and help in keeping my development on track with the NLP part of the research project.

Furthermore, a special thanks should be extended to Dr. Mohamed Firdhous, who taught the subjects of Research Methodology, Literature Reviews, and thesis writing, which formed the foundation for this research, and thank you to the examiners for their insights and their valuable comments.

I am so grateful to My colleagues who provided insight and expertise that greatly assisted the research.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents and sisters, whose love and guidance are with me in whatever I pursue. They are the ultimate role models.

Abstract

With the emergence of the Internet of Things (IoT), Smart Speakers open up a new world where we can talk to a machine for getting help in our day-to-day lives. The Smart Speaker Apps (SSA)s provide a user-friendly vocal experience to the customers by allowing them to dictate commands to the speaker through voice commands. Amazon Alexa is one of the most prevalent smart speakers which allows third-party developers to write SSAs called Skills. Due to the prevalence of Alexa, it has become vulnerable to security and privacy threats by malicious skill developers. In particular, Alexa skills could be overprivileged such that they collect more data than necessary or specified by the privacy policy in the skills description. In this research, we systematically explore skills to test whether the behaviors of the skills adhere to the privacy policy provided in the skill description. We extracted the utterances related to privacy-sensitive behavior of the skills through Natural Language Processing (NLP) techniques. Second, we implemented a dynamic testing tool Test case Generator & Invocator based on the fuzzing technique to automatically manipulate the inputs to the skills and observe the output to identify the skills which accept the privacy-sensitive information. During the study, we discovered that 21% of the tested skills accept privacy-sensitive data. We have simply focused on the real or actual behavior of the skills during the research. The claimed behavior of the skills is covered by our study, which will be the focus of further work.

Index Terms - **Alexa skills, Amazon Alexa, vulnerabilities, Internet of Things, privacy, security, Automatic testing**

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Background

Internet of Things (IoT) inspired Smart Speakers have become an essential part of smart home systems, as they provide convenience to the users to access online services and control other smart devices through voice commands. With smart speakers, users can order items online, turn on/off or check the status of smart lights or home appliances through voice commands such as “Alexa, turn on bedroom lights”. With the presence of Artificial Intelligence (AI) powered voice assistant service and the use of Natural Language Processing (NLP) techniques, smart speakers could interpret the user's voice commands in natural language and process the response in real-time. Figure 1.1 shows the components and data flow of the Alexa ecosystem. In addition to the built-in functionalities of the voice assistance services, the smart speaker service providers such as Amazon and Google allow extending their capabilities through SSAs (called *skills* by Amazon and *actions* by Google) developed by third-party developers. For instance, SSAs could support playing games or managing bank accounts. Hence, smart speakers like Amazon Alexa, and Google Assistant market today are gaining tremendous popularity than they did a few years ago. According to statistics [1], Amazon Alexa has maintained the highest market share from 2016 to 2020, and according to the latest Amazon news [2], it can control more than 100 000 devices from 9500 brands.

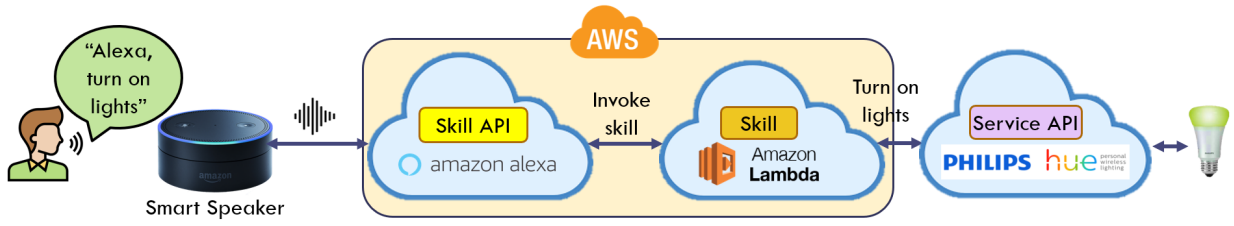

Figure 1.1: General Architecture and Data Flow To Invoke a Skill

SSAs through the integration of smart speakers and smart devices that continually capture sensitive user data, allowing the service providers and third-party developers to access private user data. This allows malicious third-party developers or service providers to gather the private user data for malicious purposes. Hence, the private data of users of the smart speakers could be vulnerable to security and privacy threats. Given the data-centric nature of Google and Amazon and the inherent ability of the smart speakers' to gather vast amounts of data without the user's knowledge, this issue aggravates. As such, with the increasing popularity of SSAs, new security implications and privacy risks have been raised. For instance, in 2018, a Portland family reported that their Amazon Alexa has recorded private conversations and sent them to a random contact in Seattle [3]. In other incidents, a toddler asked Alexa to play songs but received inappropriate adult jokes instead [4], and some healthcare workers allege that Amazon Alexa violates privacy [5]. To address the security and privacy issues in SSAs, the existing researchers have made much effort [6, 7, 8, 9, 10]. The different aspects studied in the literature include the effectiveness or the trustworthiness of the vetting process of SSAs [7, 8], and the detection of malicious SSAs which obtain sensitive information of users [6].

1.2. Challenges

During the research we have faced some challenges. Lack of available implementation or source code or execution traces of Skill is one of the disadvantages to the testers. That means, the skill is a kind of web service, which is fully black-box to the tester or the analyzer. The tester can only send inputs to the skill and observe its responses. He cannot see the inner states of the skills for the analyzing process. Therefore, the analyzer cannot determine whether the skill has been fully explored or not. Even if the skill has accepted the received input and given a valid response, it is very difficult to trigger the different behaviors of the skills without analyzing the inner states of the skill. During the research, the whole system is a black box to us.

Another challenge is that the natural language interface has been used for the Alexa conversation. The inputs and output to/from the speaker are in natural language. Alexa enables users to directly interact with various web services through natural language dialogues. In the skill-testing, the analyzer should be able to understand the responses

from the skills for giving a valid answer to the Alexa response. Although the input is the same for different skills, the received response from Alexa is quite diverse. Hence, this is also challenging to explore the behaviors of skills.

1.3. Problem Statement

Private data of Smart Speaker users could be vulnerable to security and privacy threats, due to the integration of third-party online services or smart devices with the smart speakers through Smart Speaker Apps (SSAs). In this study, we analyze the privacy behaviors of the most popular SSAs, Alexa Skills, to find whether SSAs' violate user privacy such that they extract more data than specified in the privacy policy of skill description.

1.4. Aim and Objectives

Aim: To develop an automatic tool to reduce the testing effort in identifying the Alexa Smart Speaker Apps (SSAs) which violate user privacy.

Objectives:

1. Collect skill (Alexa's SSA) descriptions from web-based Skill Stores.
2. Devise a technique to automatically extract the developer-defined privacy policy from skill descriptions and generate the utterances by using the extracted privacy policies.
3. Devise a technique to automatically invoke the skills and execute the generated utterances
4. Devise a technique to monitor the actual behavior of the skills when they are invoked
5. Detect skills that violate the defined privacy policies during the invokage stage
6. Collect the outputs from the Automated tool and Evaluate the results
7. Check the Accuracy and Performance of the implemented tool

1.5. Scope

During this research, our approach is to achieve the above mentioned objectives except for the fifth one of the project objectives. This means we test the skill's actual behavior

by giving privacy policy utterances and collecting the skills which accept the privacy-sensitive information. Hence we are not going to focus on the ‘Detect skills which violate the defined privacy policies during the invokage stage’.

Chapter 2

2. Literature Review

2.1. Introduction

This section discusses the research related to other researchers' works.

2.2. Related Works

With the exposure of security incidents, researchers have focused on analyzing security and privacy of Alexa Skills. Recently, Lentzsch et. al [7] have performed the first large scale analysis of Alexa including a dataset of 90,194 unique skills, to analyze the skill vetting process. Through their evaluation, those who have discovered a variety of flaws in the existing ecosystem that an attacker could use to debut additional attacks. These include the registration of arbitrary developer names, evading permission APIs, and changing the backend code after receiving permission to awaken latent intentions. In their future works, they have mentioned how fully automated testing of these skills is a challenging problem. In the same year, Cheng et. al [8] have also done a study on the trustworthiness of the skill certification/vetting process of Amazon Alexa and Google Assistant platforms. Their main focus was to analyze the trustworthiness of the vetting process in terms of catching policy violations by the third-party skills, detecting whether there are already existing skills that violate privacy, and studying the users' perspectives on the certification process. They have found that 100% of Alexa skills and 39% of Google actions that violate privacy policies could pass the certification out of 234 Alexa skills and 381 Google actions. However, in this study, they have synthesized privacy-violating SSAs for testing purposes. The authors have also mentioned that "Deploying automated skill-testing tools for policy violation detection" is a problem that is yet to be addressed.

To detect skills that are potentially developed by malicious developers, Guo et. al [6] have performed the first systematic exploration of interaction behaviors of skills by developing an interactive tool called "Skill Explorer". The tool may engage with skills by doing utterance extracting, question interpretation, and response production. With this tool, the authors have discovered that more than 1000 skills request users' private

information without following the developer's specifications. In our study, we planned to observe the actual privacy behavior of skills by automatically generating skill-invoking test cases by mutating the dynamic content of the utterances with privacy-related words.

Liao et. al [9] have addressed measuring the effectiveness of developer privacy policies provided in the skill description in terms of informativeness and trustworthiness. To comprehensively assess the efficacy of privacy policies offered by voice-app developers on two popular Voice Assistants systems, the researchers have carried out a large-scale data analysis. They have revealed that there are a considerable number of troublesome privacy policies and vast Internet-based platforms like Amazon, and Google is breaking their own privacy policy rules. In our study, we are planning to automatically extract the specific information on privacy policy related to data collection and sharing, to obtain the developer-defined or the expected privacy behavior of the skill.

Young, J, [21] have addressed the same problem and they have designed and developed 'SKILL DETECTIVE', an interactive testing tool capable of exploring voice apps behaviors and identifying possible policy violations in an automated manner. 54,055 nos of Amazon Alexa skills and 5,583 Google Assistant actions have been tested using this tool and found that the 6,079 skills and 175 actions are potentially violating at least one policy requirement. Apart from that, they have identified that 590 skills and 24 actions potentially violate more than one policy. They have focused not only on the textual outputs but also on unique audio files and the unique images from voice-app interactions.

As discussed in the challenges, We cannot access the implementations of the skills in the Alexa ecosystem. For analyzing the behaviors of the skills, we need to extract the specifications from the skills in its system implementation. This BlackBox testing method is based on software requirements and specifications. However, when the full system software is unavailable, deriving a specification from the implementation is challenging. During the research we use BlackBox analysis techniques, to extract the specification from the implementations of the skills. Blackbox analysis can be used to

extract system behaviors when the source code or binaries for a program are unavailable. The purpose of BlackBox analysis is to infer the behavior of a system based on its inputs and outputs. The test cases generated for the BlackBox testing are independent of the internal structure of the software system.

Blackbox analysis employs dynamic testing and fuzzing techniques. Blackbox fuzzing generates inputs without any knowledge of the program. Blackbox analysis has been used in the literature to extract abstract protocol specifications and assess security. For example, Wang et al. [33] analyzed the information flow of sensitive data using sequence diagrams illustrating the Single Sign On (SSO) protocol. Another example is the automated inference of an approximation model of e-commerce web applications from network traces (by Pellegrino et al. [34]). De et al. [35] have also utilized BlackBox testing to deduce state machines describing the TLS protocol from client and server implementations. Blackbox fuzzing has two primary subtypes: mutational and generational. The operation in mutational BlackBox fuzzing is to mutate existing data samples to create test data. And for the generation-based fuzzes, which define new test data based on models of input. AutoForge[37] and AuthScope[38] have done a hybrid fuzzing study in the context of mobile applications that communicate with cloud servers to uncover insecure access control or authorizations in the online server's implementation.

In conclusion, the researchers have proved that Alexa skills could have malicious skill behaviors due to intentionally written overprivileged skills by malicious developers or unintentional overprivileged skills due to design flaws of the platform. Hence, it is very necessary to deploy an automated testing tool to identify the privacy behavior of smart speakers' skills to check whether the skills are over-privileged such that sensitive information can be leaked violating the developer privacy policy. Hence, as a complement to existing studies, we planned to implement an automatic tool that performs BlackBox fuzzing to observe the actual privacy behavior of the skills and compare them against the developer-defined privacy policies, to detect skills violating user privacy.

2.3. Alexa Skills

Alexa skills are similar to applications. Alexa's interactive speech interface allows consumers to engage with your skill without having to use their hands. That means Alexa skills are Amazon's voice assistant applications that let users do everything from acquiring particular information to playing games and connecting smart home devices. The Alexa shop is buzzing with skills for recipes, kids, news, music, and more, much like your smartphone app store.

The first Alexa system to receive data is Automatic Speech Recognition (ASR). Alexa can recognize the voice by Automated Speech Recognition(ASR). The technique of ASR transforms spoken words into text. In a nutshell, it's the first step toward allowing speech systems like Amazon Alexa to react when we ask, "Alexa, how's the weather outside?" Voice technology can identify spoken sounds and recognize them as words using ASR. Alexa has the capability of Natural Language Understanding. The cornerstone of Alexa's strength is data and machine learning, and it is just becoming stronger as its popularity and the quantity of data it collects grow. Every time Alexa misinterprets your request, that data is utilized to make the system smarter the next time[39].

2.3.1. Publishing a third-party skill

When a developer releases a skill, he must adhere to the rules and regulations provided by the market (e.g., Amazon or Google), which is similarly related to releasing smartphone apps. The skill invocation name, cloud-based service, intents, and sample utterances are some of the essential information that should be provided in order to publish a skill on Amazon. Apart from this essential information, the developer must adhere to various market constraints. If a skill requires user privacy-related information, it should include a link to its privacy policy[22]. Markets have their own set of privacy rules that specify the types of data acquired from users, as well as how to use and distribute it. If a skill wishes to get users' information like their name, phone number, email, home address, and so on, Amazon requires that it provide a link to the skill's privacy policy throughout the development process[10]. It also needs to set permissions so that when users enable this skill, they may accept or disagree with the skill's request

for such information [23]. Markets will scrutinize such completion before distributing the skill to the skill Ecosystem as public[24].

Specifically, utterance extraction is used to set up a target skill's initial message. Because the skill does not pose a question at this point, more information should be provided in order to create a valid response. It will feed back the output when the initial feed is created and provided to the skill.

Users can engage with their services through skills, which need the transmission of data. This might include data from the user's Amazon account, such as the user's location, mobile number, email address, name, and address [25], sensitive data inferred from the user's interactions with the skills, or personal data gathered by the skills during their interactions with the users.

2.3.2. Invocation Name

a term for the invocation which describes the talent. To start a discussion with the skill, use this name. Universal uniqueness is not necessary for invocation names. Alexa offers recommendations for choosing invocation names.

2.3.3. Invoking Alexa Skills

Although skills and mobile applications are extremely similar, they have significant variances. Amazon cloud servers host the Alexa skills. As a result, customer are not required to download any binary files or conduct any installation processes. A major difference is a method of requesting services: skills need voice commands, whereas mobile applications require click activities. Users do not need to install skills on smart speakers, which is a further advantage. Consumers simply have to activate a skill in their Amazon account to utilize it[31].

Skills can be invoked in one of two ways: overtly or implicitly. When a user requests a skill by invocation name from the skill, for example, saying ”Alexa, open Ridgewood Bank” to Alexa activates the Ridgewood Bank skill, which allows users to make payments or check bank account balances. On Alexa, this sort of skill is referred to as custom skills[15]. When a user invokes an Alexa app to do something without explicitly asking for it, this is known as implicit invocation. "Hey Alexa, will it rain tomorrow?"

for example, will prompt the Weather skill to provide a weather prediction. When the interaction with the user falls into the context judged acceptable for the skill, Alexa automatically recognizes and activates the skill. That means a skill request, which does not have to include the skill invocation name, can be spoken to Alexa. Alexa can process the request and choose the best suitable skill to complete it. If the user has not yet enabled the desired ability, it may be auto-activated for them.

After publishing a skill, customers can find the published skill via a web browser or a companion app, and it can be activated by saying its invocation name into a smart-speaker. Additionally, one can learn skills via news, commercials, online discussion boards, and certain other resources[11]. Skills may be automatically detected (based on the user's voice command) and transparently activated through an IoT device, unlike smartphone applications or website plugins that must be loaded manually.

2.4. Alexa Developer Console

Alexa Developer Console is a suite of tools and services that enable a developer to build, test, and publish the skill apps in the Alexa Skill Store. To build Alexa skills, developer requires an Amazon developer account. The developer account grants the developer access to the Alexa Skills Kit (ASK) software and resources for developing and publishing Alexa skills[30].

The Alexa Skills Kit (ASK) is a platform for developing softwares that allows developers to create skills or content for Alexa. Alexa skills are similar to applications. Alexa's interactive speech interface allows consumers to engage with your skill without using their hands. Users may use their voice to do things like check the news, listen to music or play a game. Users may also operate cloud-connected gadgets with their voices. Users may, for example, ask Alexa to switch on lights or modify the thermostat. Skills are accessible on Alexa-enabled devices such as the Amazon Echo and Amazon Fire TV, as well as Alexa-enabled devices manufactured by other companies[28].

2.5. How is skill app accessed by a user?

Customer may access a skill's content by requesting Alexa to initiate the skill app via invocation name. Alexa is always willing to study unique capabilities. The voice is sent to the cloud-based Alexa system whenever a user speaks to an Alexa-enabled device and says the voice command "Alexa." The cloud-based speech service that powers Amazon's and other manufacturers' Alexa-enabled products. Developers may extend Alexa's capabilities by establishing their own cloud-based service that takes Alexa queries and gives results. Alexa detects the user's voice, decides what the user wants, and then sends a request to the skill capable of fulfilling the request. The Alexa handles voice recognition and natural language processing. Your skill is delivered as a service via a cloud platform. Alexa talks with user skill over an HTTPS request-response method. An Alexa skill returns a POST request with a JSON body whenever a user uses it.[28].

Chapter 3

3. Approach

3.1. Introduction

This third chapter includes a detailed description of the proposed method for the research.

3.2. Proposed Approach

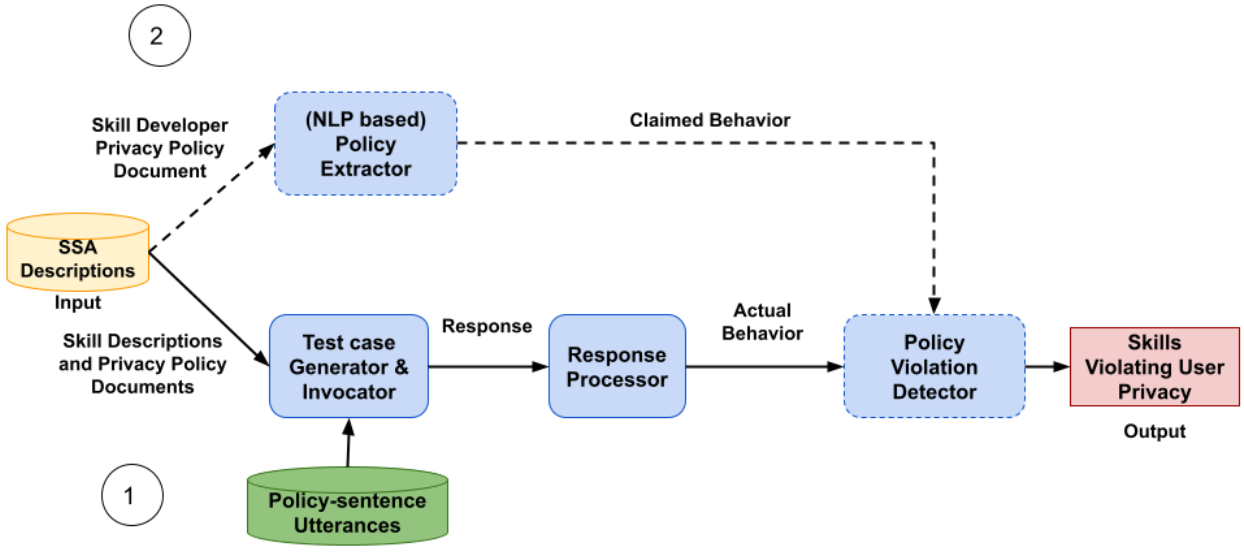

Figure 3.1: High-level Architectural Diagram of the Approach

To address the problem of analyzing the privacy of Alexa Skills, we have two main challenges. First, the implementations of the Skill Apps (skills) are not available for analysis. Hence, we cannot perform static analysis on them. Second, the skills are executed in the cloud servers. Hence, we are not able to observe the dynamic behavior by obtaining execution logs of skill executions. Therefore, we used a black-box fuzzing approach to analyze the skills based on the inputs (voice commands in natural language) given to the skill and observe the output (voice response in natural language) provided by Alexa.

The suggested approach's high-level design is depicted in Figure 3.1. It consists of two branches;

3.2.1. Testing actual behavior of the skills

Our approach takes the descriptions of the skills that we test as input, and then outputs which Alexa skills accept the sensitive information. It has three components, i.e., NLP-based Policy-sentence Utterances generator, Test case generator & Skill Invocator, and Response Processor. To prepare the input skill description corpus, we used previous research data which has used a crawler that crawls the Alexa "Skill Store" web client which includes descriptions of the skills. The skill descriptions include sample utterances and developer privacy policy URLs. In the below, we brief the task of each component.

NLP-based Policy-sentence Utterances generator:

Since the previous approaches have not been focused on random utterance generating, this is a bit challenging. Because the privacy policy text has a set of words. From this, we need to identify the privacy-sensitive words to train the model. For that, we have manually created a seed word file based on the GDPR(General Data Protection Regulation)[40], PII (personally identifiable information)[29], and other external resources. A larger set of privacy policy links are used to generate the utterances.

This component takes the URLs of the "developer privacy policies" from the skill description (in natural language) as input, and outputs the learned privacy policy utterances. We generate the random utterances using this component. This component's functions include learning privacy-related policies, generating new random utterances(which are not in the structured format of the sample utterances in the skill descriptions) from the extracted privacy policy text, and labeling the utterances as ‘Sensitive’ or ‘Not Sensitive’ by using a trained text classification model.

Test case generator and Skill invocator:

There are two primary elements to this component. Test case generator that creates more test cases to invoke Skills using the generated utterances by the "Policy-sentence

Utterances generator" and Skill invocator is used to execute the test cases by invoking the skills.

When deciding the input for this component, was a bit challenging. Every Alexa skill has a set of sample utterances mentioned in its skill description. That means the skills have predefined structured utterances for their functionalities. Apart from the mentioned functions, we need to check the behavior of skills with privacy-sensitive utterances. But, the skills are fully BlackBox to us. So, we can use one of the fuzzing techniques as mutation-based and randomly generated inputs. Previous researchers have been focused on the mutation technique to test Alexa skills. They have used the sample utterances mentioned in the skill description. But what we identified was that Alexa has the always learning capability. If we input an utterance that has the same format as the sample utterance, the output of the system will be biased according to the Alexa responses received for sent sample utterances. Hence, we used the random fuzzing technique to test the skill behaviors.

In the Alexa platform, there are several types of skill invocating methods as using Alexa SMAPI (Alexa Skill Management API), using ASK CLI(Alexa Skills Kit Command Line Interface), using Alexa SDK(Alexa Software Development), and using Alexa Developer Console[16, 17] which have been introduced for developers to test their own skills within the Alexa platform. But our research is focusing on testing the other third-party skills. Not our own skills. The Alexa Developer console is the only invocation method to test the third-party skills also.

So, in our approach, the skill invocator uses the Alexa Developer console[27] to invoke the skills by executing the test cases generated by the "Test case generator" component. When the skills are executed, this component monitors the behavior of the Skills through the received response from the Alexa Voice Assistant Service. It outputs the observed Alexa response behaviors for the executed utterances.

The input for the Test case generator is the list of skills and the generated utterances by the "Policy-sentence Utterances generator". The output of the test case generator will be the set of test cases for listed skills.

Response Processor:

The response processor is the third component that functions to analyze the Alexa responses for the requested utterances.

Within the response processor, Alexa responses should be analyzed in order to identify the Alexa accepted responses. But Alexa is an SSA who has the ability to learn within the conversations. Although we send the same utterance, the response from Alexa will be varied from time to time. That is the nature of the Alexa responses.

In our approach, we collected 1650 nos of Alexa responses by running 1300 skills with different user-sensitive information-related utterances. Those responses are manually analyzed by referring to the keywords and using additional information from sources. The responses were labeled and it is used as the seed file for the response processor. Then, a set of rules are written based on the response seed file to detect whether the invoked skill is accepted by the user's privacy-sensitive utterances.

However, The input for this Response processor is the Alexa response collected from the previously described component: Test case generator & skill invocator and the response processor checks whether the invoked skill accepts the user-sensitive information. The output of this component is whether the invoked skill is accepting the user's sensitive information or not.

3.2.2. Testing Claimed behavior of the skills

NLP Based Policy Extractor

This component takes the "developer privacy policy" from the skill description (in natural language) as input, and outputs the learned privacy policy. A text-trained model will be used to determine which privacy-sensitive data is collected, how the obtained data is used, and how long the collected sensitive data will be kept within the system.

Policy violation detector:

This component compares the defined privacy policy as learned by the "NLP-based policy extractor" component with the observed privacy-related behaviors by the skills output by the "Skill invocator and behavior monitor" component. A set of rules will be written to detect privacy violations based on domain knowledge.

After achieving both branch approaches, the two branch components are merged in order to get the *output*: *Skills violating the user privacy.*

Scope of the research: During this research, we focused on the approach described in section 3.2.1 as shown in Figure 3.1.

Chapter 4

4. Technology adapted

4.1. Introduction

This chapter aims to introduce the technology stack which has been used in the implementation stage.

4.2. Technology adapted

To implement the proposed approach, we used the Python and Java language.

In particular, to implement the Test case generator and Skill invocator we used Java as the programming language with Maven build automation tool and libraries with Cucumber and Selenium web driver.

Cucumber is a software application that facilitates Behavior-Driven Development(BDD). The Gherkin ordinary language parser is crucial to the Cucumber BDD technique. It enables the specification of expected software behaviors in a logical language and allows us to write test cases that everyone, regardless of technical skills, can understand.

Selenium WebDriver is a web framework for running cross-browser testing. This technology is used to automate web-based application testing in order to ensure that it functions as planned. To write test scripts, Selenium WebDriver can be used with a selected programming language like Java, python…etc.

Next, we implemented the "NLP-based Policy-sentence Utterances generator" using python programming language and Natural Language Processing (NLP) based techniques by using existing Python NLP libraries such as NLTK & SKlearn.

Natural Language Processing, or NLP, is a field of study that examines how computers can interpret natural language text or voice in order to do useful tasks. In order to provide the necessary tools and techniques, NLP researchers want to comprehend how humans interpret and use language. The goal of these tools will be to allow computers to alter natural languages in order to fulfill desired tasks. NLP is based on a number of fields, including computer and information sciences, linguistics, artificial intelligence, robotics, and psychology. Different disciplines of research, such as machine translation and natural language text processing, are included in NLP applications. SKlearn is a

fantastic open-source natural language processing package that is widely utilized by data scientists. It comes with a plethora of algorithms for creating machine learning models. It comes with a great documentation that assists data scientists and makes learning easy. Sci-kit Learn's key benefit is that it provides excellent intuitive class methods. It has a number of functions for converting text to numerical vectors in a bag-of-words format. It does, however, have significant drawbacks. There are no neural networks for text preparation included. If you need to do more complicated preprocessing, such as POS tagging for text corpora, you should utilize alternative NLP packages.

Text classification is a key activity in NLP, and it is based on text classifiers that evaluate text and give it a set of tags or categories based on its content. Because it classifies a document into a preset category, the task is also known as document classification.

To extract the privacy policy texts, BeautifulSoup is used and which is a Python package for parsing HTML and XML files and extracting data. It integrates with developers' preferred parser to offer fluent navigation, search, and modification of the parse tree.

The Response Processor is implemented with Java and related libraries like Apache POI. The Apache POI(Poor Obfuscation Implementation) is a collection of numerous java libraries that give various interfaces and classes via which we may manage files in our application.

Chapter 5

5. Implementation

5.1. Introduction

During this chapter, the implementation of the proposed approach will be described in detail.

5.2. Implementation

The implementation of the proposed approach (Figure 3.1) will describe in detail. The smart speaker app descriptions will be the implemented system's input, and the skills that breach the user's privacy will be the output of our implemented system.

As the first step, Alexa skill descriptions were taken. We have collected a dataset for skill descriptions from [12] Who has created a crawl to collect skills from Amazon and Google marketplaces and then created to investigate how these skills behave. Each skill's title, invocation name, needed permissions, links to privacy regulations, ratings, and other features are included in the dataset across seven countries: United States (US), United Kingdom (UK), Australia (AU), Canada (CA), Germany (DE), Japan (JP) and France (FR)[13] (see Figure 5.1 for an example of a skill's home page).

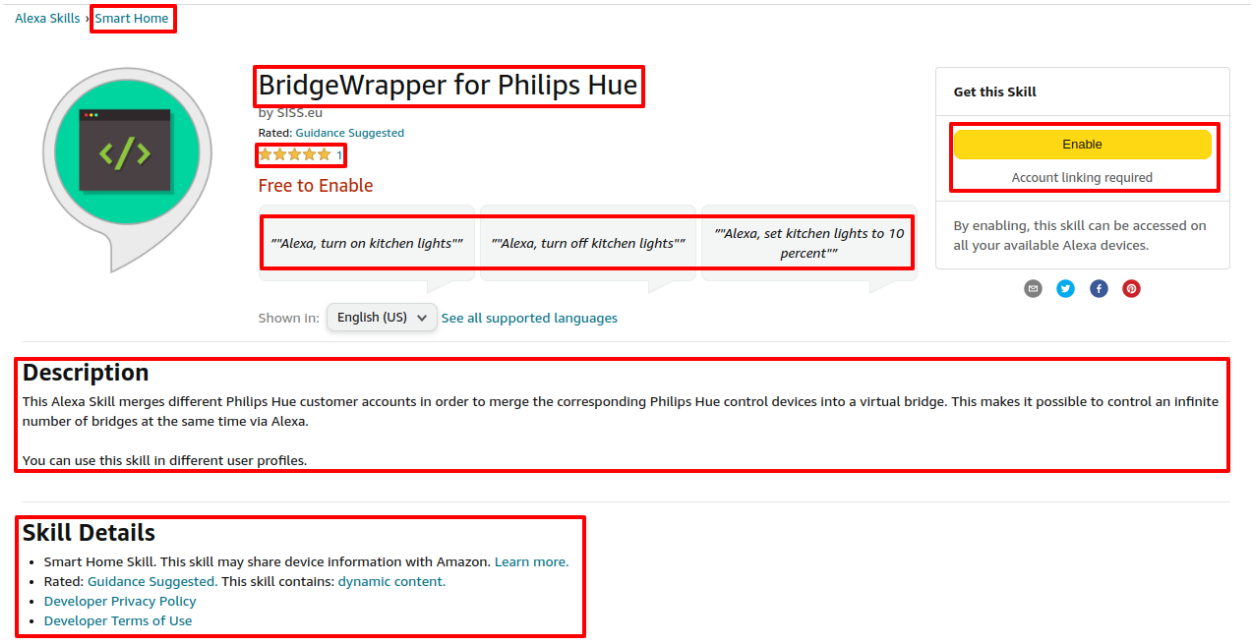

Figure 5.1 : Smart Speaker Apps Description

There are three main components of our built system: a Test case Generator & Invocator, a Policy-sentence Utterances generator (The utterances generator for the Test case generation), and a Response Processor (see the workflow in Figure 3.1).

5.2.1. Test case Generator & Invocator

The Alexa connections, requests/responses from/to the skills and looks for skill interactions in the Conversation Dataset and records them are handled by the Test case Generator & Invocator.

First, we implemented the skill invocator which is a part of a Test case Generator & Invocator. For skill invocation, we utilized the Alexa developer console[16,17]. The Alexa developer console allows dev people to test their skills during invocation and upload them for approval before making them accessible to users. During our research, this capability has been used to test the skills developed by the third-party developers. Using Selenium web driver for chrome browser, we developed web-driver based module[14] to automatically access the developer console by giving the valid credentials and by giving the skill name from the list of skills we collected from the skill description, the relevant skill can be invoked to the developer console to test.

5.2.2. Policy-sentence Utterances generator

The Policy-sentence Utterances generator will be the owner of the Utterances generated by using the skills privacy policy for the Test case generation.

Policy-sentence Utterances generator has been implemented to generate the utterances for test case generation. Our proposed approach is to test the skill by using utterances explicitly. Hence, the purpose of implementing the Policy-sentence Utterances generator is to create privacy-sensitive utterances explicitly.

The system input is the Privacy policy URLs and the output will be the generated privacy policy utterances.

Step 1: Collecting privacy policy data

As input, the list of privacy policy URLs that have been filtered off from the Alexa skill description are used.

As the first step of the process of generating utterances, the HTML text is extracted from the privacy policy URLs mentioned in the skills descriptions. For the implementation,

we used a python library called BeautifulSoup[18]. And the extracted text will be saved in a text file.

During the testing, we used 120 Nos of skill's privacy policy URLs to extract the privacy policy data.

Step 2: Preprocessing

We now have a text file containing extracted text as the input to the preprocessor that we obtained during the Data collection stage. Then the output will be a cleaned text.

The collected raw text is preprocessed using the NLTK[19] library which is a popularly used tool in the field of Natural Language Processing to remove the unwanted numbers, special characters, stopwords…etc.

Preprocessing is an essential step in achieving good outcomes with this strategy. In fact, preparation is crucial in all data-related approaches. Preprocessing is the process of converting data into an appropriate format for the procedure. It simply prepares the data so that it may be worked on. Failure to do so will have a detrimental impact on the subsequent procedure and the outcome. Preprocessing consists of four basic yet efficient steps.

I. Remove Unwanted Characters

Unwanted characters in the text, such as "1, 2, 3,..!,?...," will just add to the workload. Everything we do is based on words. It is not needed to be concerned about numbers, punctuation, and other special characteristics. Unless we are aiming to capture the phrases of the text, these unwanted characters will just slow down the process. That is not the case. We are simply searching for a label that accurately defines a group of words. As a result, we did not see any reason to keep these in mind any longer.

II. Remove Non-English Words

Some of the text we extracted from the privacy policies is not in the English Language but in another language. Since we are focusing only on the English text for the utterances generation, it is very much needed to remove the non-English words from the extracted text.

III. Remove Short Words

During the utterance generation process, the generated utterances should have meaning in full words. Hence, we removed the short words with one or two characters.

IV. Remove Stop words

The elimination of stop words is the most crucial stage. Stop words are ones that must be removed before natural language processing may begin. These are the most often used terms in the vocabulary. A stop word will just slow down the process because it does not provide much information. A stop word might be "the," "is," "at," "on," and so on. As we analyze each and every word or symbol, including them in the list will be of no value and will even be detrimental to the subsequent procedure. As a result, deleting them will be really beneficial.

So, in this preprocessing, we do the above-mentioned actions in order to improve the efficiency of our process.

Step 3: Processing

The input for this step is the cleaned text received as the output from the preprocessor. And the output will be the trained text classification model with high accuracy.

I. Breaking into phrases

As the initial step of the processing, the preprocessed text is broken into a list of words and which is used to create a phrase list by using the feature extraction function of the SKlearn of scikit[20] library, also a popularly used tool in the field of Natural Language Processing. The structural distinctions of each phrase are utilized to create a parts-of-speech (PoS) signature that is used to discriminate across types during classification. Tokens (words) are labeled using PoS tagging, which indicates if they are nouns, verbs, adjectives, and so on. A tag is assigned to each token in a phrase. The PoS tags are extracted from the phrases list using the SKlearn program. For example, if we evaluate the phrase "Read Database," we would add a VB tag to the verb "Read" The subsequent tokens in the statement would likewise be tagged, forming a pattern. The tagged phrases with VB and NN were filtered out accordingly to limit the utterances with the phrases starting with a verb and containing a noun as well.

II. Text Labeling

After generating the phrases, a seed word file with 275 nos of words has been created by manually creating a set of user-sensitive nouns and user-sensitive actions referring to

the Alexa sample utterances in the collected skill descriptions.

The filtered phrases will be labeled using the seed word file. If the utterance contains a seed word, it will be labeled as 'Sensitive.' Else, It will be labeled as 'Not Sensitive'. The labeled data will be saved in a file.

Now there is a file with labeled utterances as ‘Sensitive’ or ‘Not Sensitive’. The sample labeled utterances are shown in Table 5.1

Table 5.1 Labeled utterances

Utterance	**Label**
confirm credit card	Sensitive
identify personal information	Sensitive
include password birthday	Sensitive
send telephone number	Sensitive
Share personal information	Sensitive
delete server	Sensitive
qualify product testing	Not Sensitive
learning Alexa skill	Not Sensitive
game experience	Not Sensitive
play song	Not Sensitive

III. Text Classification

For text classification, we use the input as the created file with the labeled phrases. We can get a trained text model as the output.

The practice of classifying a text into a set of terms is known as text classification. Text classification uses natural language processing (NLP) to evaluate text and assign a set of predetermined tags or categories depending on its context. Sentiment analysis, topic recognition, and language detection are all done with NLP.

Now we have a labeled set of phrases. Then we have applied a text classification to this. The semi-supervised technique has been applied to create the basic text classification

model for the extracted phrases from the skills privacy policies.

First, the labeled text is split into two data sets the Train data and Test data in a useful manner.

- **Train Test Split**

from sklearn.model_selection, we import the train_test_split function. The train_test_split function of the sklearn.model_selection package in Python splits arrays or matrices into random subsets for train and test data, respectively.

Here we have split the total number of utterances as a train set with 41044 nos of Utterances and 10262 nos of test utterances.

- **CountVectorizer**

Here, the words of the phrases were transformed into numbers using the TF-IDF word embedding technique. That means, we have figured out how to extract information from the text in the form of word sequences, we need to figure out how to turn those word sequences into numerical characteristics, which is where vectorization comes in.

Generate a token count matrix out of a group of text documents. A sparse representation of the counts is produced by this solution. The number of features will be equal to the vocabulary size discovered by studying the data if users do not give an a-priori lexicon and people do not utilize an analyzer that does any sort of selecting features.

- **TfidfTransformer**

Tf-IDF

Term frequency-inverse Document frequency is abbreviated as tf-idf. This is commonly utilized in text mining and retrieval of information. This is a really powerful technique.

Tf-IDF considers all of the words in a collection of documents and calculates the relevance of each word in relation to that text. Simply expressed, it indicates the significance of a certain term. As a result, if the term is widely used and appears in a large number of publications, this value will be close to zero. Otherwise, it will be close to 1.

Tf-IDF is discovered for every word that passes through the preprocessing stage. To acquire a better understanding;

Term frequency:

This represents the frequency with which a specific phrase appears in a document. A

word's frequency refers to how frequently it occurs in comparison to the overall number of words in the text.

Consider the following document: doc1, "This is the first document"

The phrase frequency, tf("first",doc1)= 1/5, is now used.

Because "first" appears just once in a total of 5 words, its term frequency in d1 is 1/5.

Inverse document frequency:

The amount of information provided by a word is measured in Idf. It provides answers to queries such as how common or unusual something happens. This aids in determining the relevance or value of a certain word.

Ex: Consider the following two documents;

doc1="here is the first paper,"
doc2="here is the next paper,"
idf("this",D)=log(2/2)=0 now, where D=doc1, doc2.
IDF("first",D)=log(2/1)=0.301 is similar.

To obtain tf-idf, we easily calculate the term frequency by the inverse document frequency.

Tf-idf("first",doc1,D)=tf("first",doc1) * idf("first",D)
Tf-idf("first",doc1,D)=(1/5) * 0.301
0.0602 for Tf-idf("first",doc1,D)

This TfidfVectorizer calculation can be easily done by using scikit learn which computes the tf-idfs for us. TfidfTransformer is a transform of a count matrix to a normalized tf or tf-idf representation.

- **Multinomial Naive Bayes classifier**

This is the final estimator.

The last stage in the text classification is to train a classifier with the features that were constructed in the previous phase. There are several machine learning models that may be employed to train a final model as Naive Bayes Classifier, Linear Classifier, Support Vector Machine, Bagging Models, Boosting Models, Shallow Neural Networks, Deep Neural Networks as Convolutional Neural Network (CNN), Long Short Term Modelr (LSTM), Gated Recurrent Unit (GRU), Bidirectional RNN, Recurrent Convolutional Neural Network (RCNN) and Other Variants of Deep Neural Networks …etc. During this process of training the model, Multinomial Naive Bayes has been used.

Multinomial Naive Bayes is a probabilistic learning approach commonly used in Natural Language Processing (NLP). The method predicts the tag of a text using the Bayes theorem. It estimates the likelihood of each tag for a given sample and then outputs the tag with the highest probability. The multinomial Naive Bayes classifier is suitable for classification with discrete features (eg: count of words for text classification). Typically, integer feature values are required by the multinomial distribution.

The Naive Bayes classifier is a collection of numerous methods that all have one basic principle: each feature being categorized is unrelated to any other feature. The presence or absence of one character has no effect on the presence or absence of another.

For the implementation of the naive Bayes model, we used the sklearn library with different features.

A pipeline is created for the model generation with CountVectorizer, TfidfTransformer, and Multinomial Native Bias classifier in the sklearn python library. The purpose of the pipeline is to gather some various steps that can be cross-validated together while setting various parameter values. Here we sequentially using a list of transforms and a end estimator. The pipeline's intermediary phases must be "transformed," that is, they must use fit and transform techniques. Only fit needs to be applied by the end estimator.

IV. Improve Performance of Text Classifier:

Now we have a trained model for the phrase labeling. The accuracy of the trained model can be predicted as a classification report showing the main classification metrics. We can create a classification_report which is imported from sklearn.metrics. This will return the Text summary of the precision, recall, and F1 score for each class. Dictionary returned if output_dict is True. The reported averages include macro average (averaging the unweighted mean per label), weighted average (averaging the support-weighted mean per label), and sample average (only for multilabel classification). Micro average (averaging the total true positives, false negatives, and false positives) is only shown for multi-label or multi-class with a subset of classes because it corresponds to accuracy otherwise and would be the same for all metrics.

Some modifications to the entire process step can be made to obtain high accuracy for the trained model. During this research, we have done the following things to help text

classification algorithms perform better.

1. Text Preprocessing: Text preprocessing has been done to reduce noise in text data such as stopwords, special characters, punctuation marks, stop words, and so on.
2. Use a ground truth and check the accuracy: A balanced ground-truth feed file has been created based on the sensitive nouns and actions and has been used to check the accuracy. The created ground truth file consisted of 210 nos of 'Sensitive' utterances and also 210 nos of 'Non-Sensitive' utterances. This helps to improve the accuracy of the trained model.
3. Increase the data set: We have increased the extracted privacy policy text data set and achieved more accuracy.

V. Categorize the Utterances

Next, we need to structure the labeled phrase in order to arrange it more effectively by grouping the related phrases together. we used the labeled phrase - Sensitive as the input and the output is a set of groups with labeled phrases. To group the sensitive utterances, we selected a set of keywords from a NIST (National Institute of Standards and Technology) report[29].

Table 5.2 Keywords related to Utterances category

Utterances category	**Keywords**
Personal Information	Name, Email, Birthday, Age, Gender, Location, Contact, Phonebook, Address, Zipcode, Postal code, Phone number, Passport number, Driver license number, email
Employment Related Information	Profession, company, office, office address, employment type, employment id, employee number, client, mail, email, business
Security Related Data	Username, password, login, sign in, email, signature
Financial related Data	Account, Credit, debit, purchase, bill, pay card, allowance, order, cart, money, shopping, charge

Personal services (Verb Set)	Access, post, Ask, Assign, Collect, Create, Enter, Import, Obtain, Observe, Provide, Receive, Request, Share, Use, Include, Integrate, Monitor, Process, See, Utilize, Retain, Delete, Erase, Remove, Store, Transfer, Communicate, Apply, place, Sell, Send, Update, View, Need, Require, Save, alexa
Social Media	fb, facebook, insta, instagram, twitter, telegram, messenger

5.2.3. Response Processor

The Response Processor analyzes the response from the Alexa.

After creating the Skill Invocator module, 50 Nos of privacy-sensitive utterances were executed for more than 1300 skills in 21 categories and stored the responses from the Alexa. A seed file has been created by manually analyzing the responses which are labeled as 'Accepted', 'Not Accepted', and 'Can not Say'. Here we check the response received from Alexa is available in the seed file. If it is available in the seed file, the label of the listed utterance will be used as the Skill status. Otherwise, it will be marked as 'Pending'. Then we again manually analyzed the 'Pending' skill status and added them to our response list to improve the accuracy of the Response Processor.

We have the generated Policy-sentence Utterances created by using the implemented Policy-sentence Utterances generator for creating the test cases. During the research, we focused on test case generation with only one utterance. For example, the test steps for one test case are;

1. Log in to the Alexa Developer Console with a valid Amazon account
2. Invoke the Skill
3. Store the response from the Alexa
4. Send the generated utterance to Alexa
5. Collect the response from Alexa
6. Analyze the response with the existing responses seeds file
7. Store the analyzed data as the output file

The test case execution has been done by using the selenium framework with cucumber. The execution steps for a test case are as First the user login into the system (Alexa Developer Console). Then a skill is invoked and a privacy-sensitive utterance will be

sent to the Alexa through the Alexa developer console. The response from Alexa for the sent utterances will be collected and that response will be analyzed based on the seed word file (Refer to *Appendix B* for all labeled response types) which was created previously by executing privacy-related utterances.

Example:

1. Alexa Response: '<Audio only response>'

The '<Audio only response>' is accepted by Alexa. Therefore this is made as 'Accepted'

2. Alexa Response: 'Sorry, I don't know that.'

This response means that Alexa is not aware of the sent privacy-sensitive utterance. This means the skill is not aware of that privacy-sensitive utterance. So, this is labeled as 'Not Accepted'

3. Alexa Response: 'Hmm, I found a few skills that might help.'

Here, Alexa tries to list some skills within the invoked skill. This is a bit difficult to understand whether Alexa is accepting the privacy-sensitive utterance or not. Hence, this is labeled as "Can not Say".

Example:

- **The Input for the Implemented System:**

Sample test case:

Username *: abc@gmail.com*
Password *: abc@123*
Skill category *: Lifestyle,*
Skill Id *: B07D1GT8MT,*
Skill Invocation Name *: grammar teacher,*
Privacy-related Utterance *: save personal information provided*

- **The system output:**

Skill category *: Lifestyle,*
Skill Id *: B07D1GT8MT,*

Skill Invocation Name *: grammar teacher,*
Privacy-related Utterance *: save personal information provided,*
Alexa response *: <Audio only response>,*
skillStatus *: Accepted,*
Alexa Conversation *: {*

"Request1":"alexa open grammar teacher",
"Response1.1":"Hello there! I am your grammar teacher. I will try my best to correct any mistakes you make. But remember, like all humans and machines, I too am not perfect. You can ask for help anytime by saying, ask grammar teacher for help. Is there anything I can help you with right now?",
"Request2":"save personal information provided",
"Response2.1":"<Audio only response>",
"Request3":"alexa close grammar teacher",
"Response3.1":"Do you mean ?"
}

Chapter 6

6. Evaluation

6.1. Introduction

This chapter focuses on how testing strategies carried out for the research sub question in terms of the evaluation measurements

6.2. Evaluation

We have created the text classified model to label the privacy policy-related utterances within the privacy-sensitive utterance generator component. Figure 6.1 depicts the classification report for the prediction of the test data set using the trained model.

	precision	recall	f1-score	support
Sensitive	1.00	0.83	0.91	4286
Not Sensitive	0.89	1.00	0.94	5976
accuracy			0.93	10262
macro avg	0.94	0.91	0.92	10262
weighted avg	0.94	0.93	0.93	10262

Figure 6.1 Classification report for the prediction of the test data set using the trained model

The implemented system runs for 5268 skills of 21 skill categories and found that 1219 nos of skills have been accepted for privacy-sensitive data-related utterances. The accepted skills count for privacy-sensitive utterances is 23% of the total skills count.

Research Question 1: In which category of skills is extracting the privacy data?

We observed that all skill categories allow privacy-sensitive utterances, with Music & Audio, Business & Finance, and the weather being the top skill categories that accept privacy-sensitive utterances. Figure 6.2 depicts the approved skill count for each skill category as a chart.

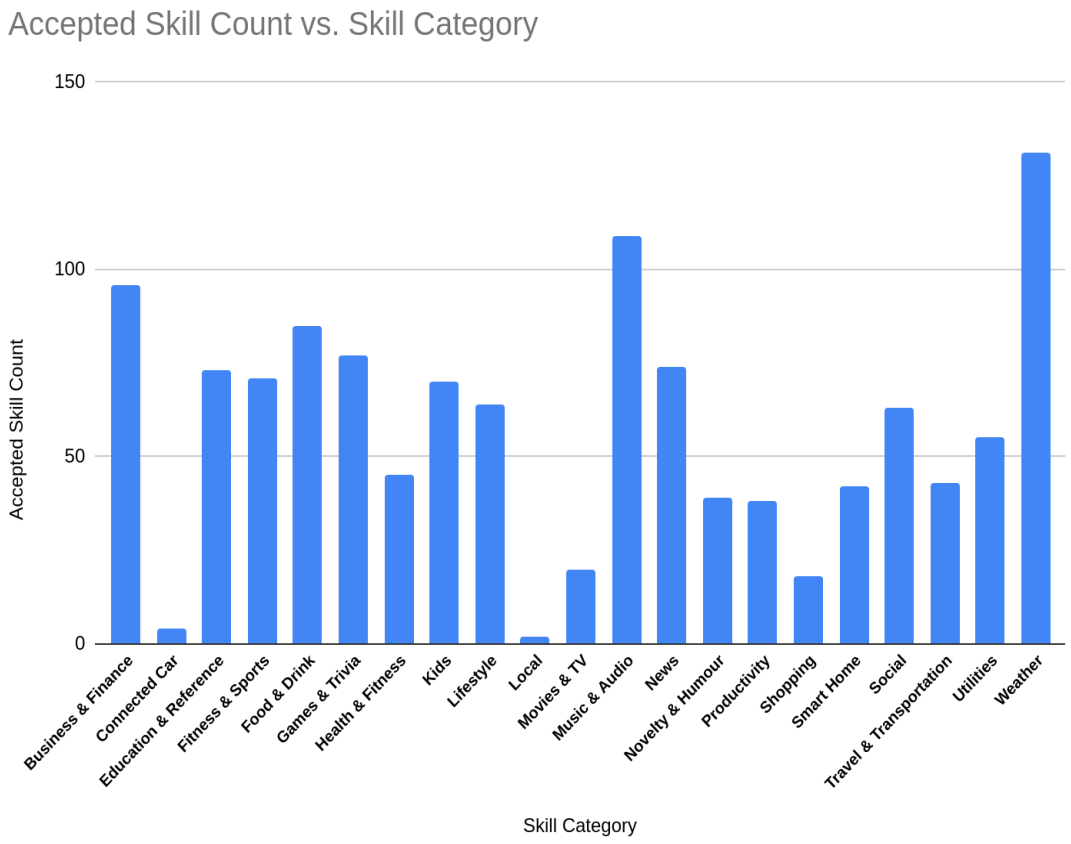

Figure 6.2 Accepted skill count vs skill Category

Research Question 2: What kind of user-sensitive data are the skills most willing to accept?

We investigated which kind of utterances are most commonly accepted by skills. We discovered that the most accepted utterances fall under the Personal Information category. Table 6.1 shows the summary of the accepted skill count for each utterance category.

Table 6.1 Utterance Category vs Accepted Skill Count

Utterance Category	Accepted Skill Count %
Personal Information	51.54
Financial related Data	30.42
security-related information	26.26
Employment Information	21.95
Personal services	18.68
Other	18.61
social media	13.33
Total Accepted skill count %	**23.14**

Research Question 3: What are the skill categories for specified utterance categories?

We discovered that the majority of the skill categories were centered on the user's personal data. Music & Audio, Fitness & Sports, and Weather are the top three skill categories for receiving personal data. The recognized skill counts for all skill categories are shown in Table 6.2.1 below.

Table 6.2.1: Utterance Category - Personal Information

Skill Category	Accepted Skill count %
Music & Audio	13.11%
Fitness & Sports	8.67%
Novelty & Humour	7.26%
Weather	7.26%
Food & Drink	7.03%
Games & Trivia	7.03%
Education & Reference	6.56%
Smart Home	5.85%
Business & Finance	5.62%
Kids	4.92%
Movies & TV	3.98%

Productivity	3.51%
Social	3.51%
News	3.28%
Lifestyle	3.04%
Utilities	3.04%
Travel & Transportation	1.87%
Health & Fitness	1.64%
Shopping	1.41%
Connected Car	0.94%
Local	0.47%

Skills accept the user security-related information as the top second utterance category (refer Table 6.2.2). We identified that Business & Finance, News, and Weather have the highest accepted skill count % related to User security-related information.

Table 6.2.2: Utterance Category - User security-related information

Skill Category	**Accepted Skill count %**
Business & Finance	11.03%
News	10.56%
Weather	8.92%
Food & Drink	8.22%
Fitness & Sports	7.51%
Utilities	7.04%
Health & Fitness	6.81%
Lifestyle	6.57%
Travel & Transportation	5.40%
Games & Trivia	4.93%
Music & Audio	4.93%
Education & Reference	4.69%
Kids	3.99%
Social	3.76%
Productivity	2.11%
Smart Home	2.11%
Movies & TV	0.47%
Novelty & Humour	0.47%
Shopping	0.47%

The below tables Table 6.2.3, Table 6.2.4, Table 6.2.5 are also related to other utterance categories.

Table 6.2.3: Utterance Category - Financial related Data

Skill Category	Accepted Skill count %
Weather	24.79%
Music & Audio	21.37%
Business & Finance	13.68%
Education & Reference	10.26%
Kids	6.84%
Games & Trivia	4.27%
Shopping	4.27%
Social	3.42%
Lifestyle	2.56%
News	2.56%
Smart Home	2.56%
Travel & Transportation	1.71%
Health & Fitness	0.85%
Utilities	0.85%

Table 6.2.4: Utterance Category - Personal services

Skill Category	Accepted Skill count %
Social	22.22%
Kids	16.67%
Education & Reference	8.89%
Lifestyle	6.67%
Utilities	5.56%
Weather	5.56%
Food & Drink	4.44%
Games & Trivia	4.44%
Health & Fitness	4.44%
Music & Audio	3.33%
News	3.33%
Shopping	3.33%
Business & Finance	2.22%
Fitness & Sports	2.22%
Smart Home	2.22%
Travel & Transportation	2.22%
Movies & TV	1.11%
Productivity	1.11%

Table 6.2.5: Utterance Category - Employment Information

Skill Category	Accepted Skill count %
Weather	20.21%
Games & Trivia	14.89%
Productivity	12.77%
Food & Drink	9.57%
Lifestyle	8.51%
Education & Reference	5.32%
Travel & Transportation	5.32%
Business & Finance	4.26%
News	4.26%
Music & Audio	3.19%
Utilities	3.19%
Kids	2.13%
Novelty & Humour	2.13%
Shopping	2.13%
Social	2.13%

Performance: Our system has analyzed 5268 skills within 179 hours (using a machine with a 3.6GHz CPU 16GB memory, 1TB hard drive, and the Ubuntu 20.04 operating system).

Results Comparison with Related work

The end goal of our research is matched with two similar works entitled SkillExplorer[6] and SkillDetective[21]. While SkillExplorer has not yet made its implementation available, the SkillDetective tool has. So, we compared the results against those of the SkillDetective tool.

Our tool was tested against the classification abilities of the SkillDetective[21] tool. Since we have completed the research only for accepting user privacy data by skill, the comparison has been done for collecting the skills which accept privacy-sensitive information. We have compared 100 skills and 89% of the Results of our tool are matched with the SkillDetective[21] tool results. The sample skill set is shown below in table 6.3.

Table 6.3: Results comparison with SkillDetective[21] tool

Skill Category	**Accepted Skill count %**
B07DFCXXM5	Say Please
B087Z4F96J	Math Whiz
B07R97WNFR	My bestie
B0837HWNY5	My burns
B07N626993	Blood donation helper
B0829RDY2T	HealthDataGatherer
B01JG73CJI	spare the air
B0746FHXLY	Adopt A Pet
B07CLN9Q8T	Sneeze Forecast
B07NGT7LG4	You Choose

Chapter 7

7. Discussion

7.1. Introduction

This section will be used to discuss the limitations of the implemented system.

7.2. Discussion

The increasing use of smart speaker devices and other voice-first devices has raised user and developer awareness of various privacy concerns. After reviewing and researching these potential issues, there appear to be various concerns about the possible exploitation and flaws.

In this research, we proposed a systematic analysis of the skill behaviors of Alexa in relation to their privacy policies. A set of NLP-based methods, such as Privacy-policy utterance generator, skill behavior understanding with Skill invocator and behavior monitor, and test case generation, policy violation detector, are fundamental strategies that enable the skill-testing.

In this thesis, we addressed the problem of research as to how we can generate privacy-sensitive utterances using privacy policies? In which category of skills are extracting the privacy data?, What kind of user-sensitive data are the skills most willing to accept? and What are the skill categories for specified utterance categories ?. In the evaluation, we have evaluated these research questions with the results.

When invoking the skill, there were certain impediments to our selected invocation techniques during the implementation of the skill invocator with behavior monitor. Some of the Alexa APIs only could be used for developers' custom skills. But, in this case, we needed to invoke skills that were hosted by third-party developers. There was only one way to invoke third-party skills, called the Alexa Developer console.

In the Privacy-policy utterance generator, current NLP tools which have been used in the implemented system have false positives (eg: the created pos tags for some verbs are NN). Another limitation is the simulator. It currently has limitations on how it interacts with mobile phones and how it transmits geographic information. And also some skills are asked for its echo device connection to continue the further process. Since we have focused on text(Non-Audio) conversations with Alexa, this is also a limitation of our

system. During the research, we used Skills in the English language. Hence the privacy policy utterances are also generated in the same English language. Therefore the implemented system has been limited to English Language skill testing. The skill-testing is a time-consuming process since it necessitates a new start for each utterance interaction each time for testing. We were unable to verify every skill with a number of utterances at maximum capacity utilizing our Slow system due to the inherent latency of skill testing. Hence, we have tested the skill with one utterance from different clusters at a time.

Conclusion

In this work, we designed and implemented a tool to identify the privacy-sensitive utterances accepting skills in current SSA platforms. Using the implemented system, we have tested the skills and identified that the 21% of the tested skills are accepting privacy-sensitive data.

Future works

We have simply focused on the real or the actual behavior of the skills during the research. Our research, which will be the subject of future work, includes the claimed behavior of the skills.

References

[1] Vailshery, L. S. (2021, April 7). *Global smart speaker market share 2020*. Statista. https://www.statista.com/statistics/792604/worldwide-smart-speaker-market-share/.

[2] Richter, F. (2020, July 23). *Infographic: The Growing Footprint of Amazon's Alexa*. Statista Infographics. https://www.statista.com/chart/22338/smart-home-devices-compatible-with-alexa/.

[3] Elise Herron | Published May 26, 2018 :*Portland Family Says Their Amazon Alexa Recorded Private Conversations and Sent Them to a Random Contact in Seattle*. Willamette Week. https://www.wweek.com/news/2018/05/26/portland-family-says-their-amazon-alexa-recorded-private-conversations-and-sent-them-to-a-random-contact-in-seattle/.

[4] Report, P. S. *Toddler asks Amazon's Alexa to play song but gets porn instead*. New York Post.https://nypost.com/2016/12/30/toddler-asks-amazons-alexa-to-play-song-but-gets-porn-instead/.

[5] www.govinfosecurity.com. (n.d.). Healthcare Workers Allege Amazon Alexa Violates Privacy https://www.govinfosecurity.com/healthcare-workers-allege-amazon-alexa-violates-privacy-a-17002

[6] Guo, Z., Lin, Z., Li, P. and Chen, K., 2020. Skillexplorer: Understanding the behavior of skills in large scale. In 29th {USENIX} Security Symposium ({USENIX} Security 20) (pp. 2649-2666).

[7] Lentzsch, C., Shah, S.J., Andow, B., Degeling, M., Das, A. and Enck, W., 2021, February. 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.

[8] Cheng, L. et al., 2020. Dangerous Skills Got Certified: Measuring the Trustworthiness of Skill Certification in Voice Personal Assistant Platforms. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*.

[9] Liao, S., Wilson, C., Cheng, L., Hu, H. and Deng, H., 2020, December. Measuring the effectiveness of privacy policies for voice assistant applications. In *Annual Computer Security Applications Conference* (pp. 856-869).

[9] Liao, S., Wilson, C., Long, C., Hu, H. and Deng, H., 2021. Problematic Privacy Policies of Voice Assistant Applications. IEEE Security & Privacy.

[10] Zhang, N., Mi, X., Feng, X., Wang, X., Tian, Y. and Qian, F. (n.d.). 2019. Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems. IEEE Symposium on Security and Privacy (SP)

[11] xie00059 (2022). Amazon-Alexa-UQ-AAS21-datasets. [online] GitHub. Available at: https://github.com/xie00059/Amazon-Alexa-UQ-AAS21-datasets [Accessed 12 Jun. 2022].

[12] alexa-skill-analysis-org (2021). A Privacy & Security Analysis of the Alexa Skill Ecosystem. [online] GitHub. Available at: https://github.com/alexa-skill-analysis-org/alexa-skill-analysis-org

[13] Lentzsch, C., Shah, S.J., Andow, B., Degeling, M., Das, A. and Enck, W. (2021). Hey Alexa, is this Skill Safe?: Taking a Closer Look at the Alexa Skill Ecosystem. Proceedings 2021 Network and Distributed System Security Symposium. [online] doi:10.14722/ndss.2021.23111.

[14] Seleniumhq.org. (2022). Selenium - Web Browser Automation. [online] Available at: https://www.seleniumhq.org [Accessed 12 May 2022].

[15] Amazon (Alexa). (n.d.). Understand How Users Invoke Custom Skills | Alexa Skills Kit. [online] Availableat:https://developer.amazon.com/en-US/docs/alexa/custom-skills/understanding-how-users-invoke-custom-skills.html [Accessed 12 Jun. 2022].

[16] Amazon (Alexa). (n.d.). Test with the Alexa Simulator | Alexa Skills Kit. [online] Available at: https://developer.amazon.com/en-US/docs/alexa/devconsole/alexa-simulator.html [Accessed 12 Jun. 2022].

[17] www.amazon.com. (n.d.). Amazon Sign-In. [online] Available at: https://developer.amazon.com/alexa/console/ask [Accessed 12 Jun. 2022].

[18] Crummy.com. (2019). Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation. [online] Available at: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[19] Nltk.org. (2019). NLTK Book. [online] Available at: https://www.nltk.org/book/.

[20] scikit-learn.org. (n.d.). Documentation scikit-learn: machine learning in Python — scikit-learn 0.18.2 documentation. [online] Available at: https://scikit-learn.org/0.18/documentation.html.

[21] Young, J., Liao, S., Cheng, L., Hu, H. and Deng, H. (n.d.). SkillDetective: Automated Policy-Violation Detection of Voice Assistant Applications in the Wild. [online] Available at: https://www.usenix.org/system/files/sec22summer_young.pdf [Accessed 13 Jun. 2022].

[22] Amazon (Alexa). (n.d.). Certification Requirements | Alexa Skills Kit. [online] Available at: https://developer.amazon.com/docs/custom-skills/certification-requirements-for-custom-skills.html/ [Accessed 13 Jun. 2022].

[23] Amazon (Alexa). (n.d.). Request Customer Contact Information for Use in Your Skill | Alexa Skills Kit. [online] Available at: https://developer.amazon.com/zh/docs/custom-skills/request-customer-contact-information-for-use-in-your-skill.html [Accessed 13 Jun. 2022].

[24] Edu, J., Ferrer Aran, X., Such, J. and Suarez-Tangil, G. (2021). SkillVet: Automated Traceability Analysis of Amazon Alexa Skills. IEEE Transactions on Dependable and Secure Computing, pp.1–1. doi:10.1109/tdsc.2021.3129116.

[25] Amazon (Alexa). (n.d.). Host a Custom Skill as a Web Service | Alexa Skills Kit. [online] Available at: https://developer.amazon.com/en-US/docs/alexa/custom-skills/host-a-custom-skill-as-a-web-service.html [Accessed 13 Jun. 2022].

[26] Zhang, N., Mi, X., Feng, X., Wang, X., Tian, Y. and Qian, F. (2019). Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems. 2019 IEEE Symposium on Security and Privacy (SP). doi:10.1109/sp.2019.00016.

[27] www.amazon.com. (n.d.). Amazon Sign-In. [online] Available at: https://developer.amazon.com/alexa/console/ask.

[28] Amazon (Alexa). (n.d.). What is the Alexa Skills Kit? | Alexa Skills Kit. [online] Available at: https://developer.amazon.com/en-US/docs/alexa/ask-overviews/what-is-the-alexa-skills-kit.html.

[29] Mccallister, E., Grance, T. and Scarfone, K. (2010). Special Publication 800-122 Guide to Protecting the Confidentiality of Personally Identifiable Information (PII) Recommendations of the National Institute of Standards and Technology. [online] Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-122.pdf.

[30] Amazon (Alexa). (n.d.). Create and Manage Skills in the Developer Console | Alexa Skills Kit. [online] Available at: https://developer.amazon.com/en-US/docs/alexa/devconsole/about-the-developer-console.html [Accessed 21 Jun. 2022].

[31] Amazon (Alexa). (n.d.). Build a Skill End-to-end Using an Alexa-hosted Skill | Alexa Skills Kit. [online] Available at: https://developer.amazon.com/en-US/docs/alexa/hosted-skills/build-a-skill-end-to-end-using-an-alexa-hosted-skill.html.

[32] Johannisson, K. (2005). Formal and informal software specifications. Göteborg Dept. Of [33] Sun, S.-T. and Beznosov, K. (2012). The Devil is in the (Implementation) Details: An Empirical Analysis of OAuth SSO Systems, pp. 378–390.

[34] G. Pellegrino and D. Balzarotti, (2014). Toward black-box detection of logic flaws in web applications. NDSS

[35] J. De Ruiter and E. Poll, (2015). Protocol state fuzzing of {TLS} implementations, in USENIX Security, pp. 193–206.

[36] Turner, R. and Angius, N. (2013). The Philosophy of Computer Science. plato.stanford.edu.

[37] Zuo, C., Wang, W., Wang, R. and Lin, Z. (2016). Automatic Forgery of Cryptographically Consistent Messages to Identify Security Vulnerabilities in Mobile Services.

[38] Zuo, C., Zhao, Q. and Lin, Z. (2017). AuthScope: Towards Automatic Discovery of Vulnerable Authorizations in Online Services. [online] doi:10.1145/3133956.3134089, pp. 799–813.
[39] Alexa Privacy and Data Handling Overview (20180720). (n.d.).
[40] GDPR (2016). General Data Protection Regulation (GDPR). [online] General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/.

Appendixes

Appendix A

IoT - Internet of Things

NLP - Natural Language Processing

SSAs - Smart Speaker Apps

AI - Artificial Intelligence

API - Application Programming Interface

HTTP - Hypertext Transfer Protocol

NLTK - Natural Language Toolkit

REST - Representational state transfer

Alexa SMAPI - Alexa Skill Management API

ASK CLI - Alexa Skills Kit Command Line Interface

Alexa SDK - Alexa Software Development

PoS - Part of Speech

ASR - Automatic speech recognition

BDD - Behavior Driven Development

CNN - Convolutional Neural Network (CNN)

LSTM - Long Short Term Modelr (LSTM)

GRU - Gated Recurrent Unit (GRU)

RNN - Bidirectional RNN

RCNN - Recurrent Convolutional Neural Network (RCNN)

GDRP - General Data Protection Regulation

PII - Personal Identifiable Information

URL - Uniform Resource Locator

HTML - HyperText Markup Language

XML - Extensible Markup Language

Appendix B - Responses List

Response_keyword	**Accepting_Privacy_Data**
Blank response	Accepted
<Audio only response>	Accepted
<Short audio>	Accepted
OK	Accepted
Hello,	Accepted
Sorry, I am unable to fulfill your request on this device. Please try again from your echo device.	Accepted
Go on?	Accepted
Say yes to continue	Accepted
I'm sorry I didn't get that. Do you want to hear	Not Accepted
Sorry, I didn't understand that.	Not Accepted
Sorry, I'm not sure.	Not Accepted
Sorry, I don't know that.	Not Accepted
Hmm, I don't know that one.	Not Accepted
Sorry, I'm not sure about that.	Not Accepted
What would you like to know?	Not Accepted
Hmm, I don't know that.	Not Accepted
Sorry. I did not understand that. Would you like to learn more?	Not Accepted
Looks like I don't have that information yet.	Not Accepted
Sorry, I don't know that one.	Not Accepted
Hmm, I'm not sure.	Not Accepted
Sorry, I don't know about that. Please try again.	Not Accepted
Please try again or say help for a list of possible commands	Not Accepted
Sorry, I don't know about that. Please try again.	Not Accepted
Not a valid code	Not Accepted
getaddrinfo ENOTFOUND odata.intel.com odata.intel.com:443	Not Accepted

Sorry, this is not a valid command. Please say, Help, to hear what you can say.	Not Accepted
Sorry, that's an invalid	Not Accepted
I'm sorry, I currently do not know the list for com. What else can I help with?	Not Accepted
Sorry I did not understand your answer.	Not Accepted
Sorry. I don't know that...	Not Accepted
Sorry, I could not find...	Not Accepted
I'm not sure what you just said. Please say "help" for voice operation guide.	Not Accepted
I'm sorry, you must specify a valid bus stop. Goodbye	Not Accepted
Unfortunately I did not get all needed input. Can I do something else for you?	Not Accepted
I did not understand that response...	Not Accepted
I'm sorry, I didn't get that. Please try again.	Not Accepted
That answer is wrong	Not Accepted
Sorry, I'm having trouble accessing	Cannot Say
I'm sorry, I currently do not know	Not Accepted
Response is not related to request	Cannot Say
Hmm, I don't have an answer for that.	Not Accepted
Sorry, I don't have an answer for that.	Not Accepted
Sorry I couldn't understand, but I may have a few recommendations.	Cannot Say
Hmm, I found a few skills that might help.	Cannot Say
I can't do that on iphone. Would you like to use your THIRD PARTY AVS OVERRIDE?	Cannot Say
Here's something I found on the web	Cannot Say
We are not able to complete the program at this time. We apologize for this inconvenience. Please try again later.	Cannot Say
Sorry, I don't have an answer for that.	Not Accepted
Do you mean	Cannot Say
Did you mean	Cannot Say

Please hold while I symlink you to device	Accepted
Okay. Before I can tell you about your	Accepted
I didn't understand	Not Accepted
That answer is wrong.	Not Accepted
Do you want to listen another fact. say yes or no.	Not Accepted
I'm sorry, I currently do not know that.	Not Accepted
I don't know that one.	Not Accepted
I don't know that.	Not Accepted
I can't find any	Not Accepted
Do you want another fact. Say yes or No.	Not Accepted
I didn't understand what you meant	Not Accepted
does not match any job in my database	Not Accepted
I am unsure of the answer at this time	Not Accepted
Sorry. I did not understand that.	Not Accepted
Sorry. I could not find item	Not Accepted
I am afraid that I do not understand	Not Accepted
is not supported.	Not Accepted
How about trying a	Cannot Say
try again with different	Not Accepted
There are no	Not Accepted
not supported on this device.	Not Accepted