CCS '23 Artifact Appendix: SkillScanner: Detecting Policy-Violating Voice Applications Through Static Analysis at the Development Phase

Song Liao

A Artifact Appendix

A.1 Abstract

In this work, we design and develop SkillScanner, an efficient static code analysis tool to facilitate third-party developers to detect policy violations early in the skill development lifecycle. To evaluate the performance of SkillScanner, we conducted an empirical study on 2,451 open source skills collected from GitHub. SkillScanner effectively identified 1,328 different policy violations from 786 skills.

A.2 Description & Requirements

A.2.1 Security, privacy and ethical concerns

Due to the copyright, we didn't provide skill code from other developers but provided a document with links to Alexa skills. You can download skill code from other repositories, as shown in https://github.com/CUSecLab/SkillScanner/blob/main/skills_code/all_skills_dataset.csv. You can download the skill code from GitHub and perform the analysis using SkillScanner.

A.2.2 How to access

Everyone can download the SkillScanner from the GitHub repository: https://github.com/CUSecLab/SkillScanner.

A.2.3 Hardware dependencies

The hardware required for SkillScanner depends on the code size to be analyzed. Normally, skills have less than 100K lines of code, so SkillScanner requires at least 2 cores CPU and 8 GB RAM.

A.2.4 Software dependencies

One of the most important packages is CodeQL and any user planning to use SkillScanner needs to download CodeQL first for running taint analysis for skills. Users can download CodeQL from https://github.com/github/codeql-action/releases (also listed in readme.md file). For other Python libraries, you can install them using requirements.txt, such as "pip install -r requirements.txt".

If you want to scan/download skill code datasets from GitHub or analyze skill content/html safety, you need to apply for tokens about GitHub, Google Perspective, and Virustotal. Then you need to put them in the "tokens.txt" file in the "skillscanner" folder.

A.2.5 Benchmarks

We have provided our dataset in https://github.com/CUSecLab/SkillScanner/blob/main/skills_code/all_skills_dataset.csv. However, due to ethical concerns, we couldn't provide the downloaded repositories to others. Users who are interested in the dataset can run the "search_github.py" and ""clone_repo.py" codes to search and download code from GitHub.

A.3 Set-up

A.3.1 Installation

You need to download the CodeQL from CodeQL for the skill taint analysis. After downloading and unzipping it, rename it as "codeql-home" and put it in the root path of this repo. You also need to install Python libraries using "pip install -r requirements.txt".

A.3.2 Basic Test

When you plan to scan a skill, go to the "skillscanner" folder and run with: "python scan_skills.py ../skills_code 1". "1" means there might be several skills in the target folder and "0" means only one skill. Ensure that all the skill files are in one folder. The results will be in the folder "skillscanner/results" and each skill will have a folder for storing results.

A.4 Evaluation workflow

A.4.1 Major Claims

SkillScanner is able to scan the skill code in a folder and provides a report for reporting all potential violations in the skill. The average time for scanning one skill is 76 seconds, and normally it should not exceed 120 seconds.

A.4.2 Experiments

(E1): If you download the skill code from https://github.com/3unyt/Alexa-Intern-Helper and analyze it with SkillScanner (run it with "python scan_skills.py ../{target_folder} 1"), you will get a report (in the folder "skillscanner/results/{skillname}") with the following content:

```
Scanning the skill cost: 48.34189486503601s.
The intent number is: 2
The slot number is: 4
The function number is: 13
The sample number is: 9

Data collection in the skill code:
outputs data collection /home/song/rsc/ccs_skill_scanner_test/
  skillscanner/skills_code/3unyt/Alexa-Intern-Helper/lambda/
  index.js        hello! welcome to amazon intern helper. what is
  your name?      collect data name
permission data collection      name
permission data collection      email

Issues in the skill code:
This skill has an incomplete privacy policy: data ['name']
  collected in output is not mentioned in privacy policy.
This skill has an incomplete privacy policy: data ['name',
  'email'] collected in permission is not mentioned in privacy
  policy.
This skill has an incomplete/lacks a privacy policy: data ['used
  in database'] collected and stored in slots is not mentioned in
  privacy policy.
This skill has an incomplete/lacks a privacy policy: data ['used
  in database'] collected and stored in permissions is not mentioned
  in privacy policy.
permissions asked not used      name
```

An intent represents an action that fulfills a user's spoken request. Intents normally have two values: the intent name and sample utterances.

Sample utterances are a set of likely spoken phrases mapped to the intents and developers should include representative phrases so that the interaction model can better learn the sentence pattern.

In addition, intents can optionally have arguments called slots. Slot is the variable that can capture a specific type of user's verbal reply, such as username or user address. More details can be found in our paper in Section 2.1.

A.5 Notes on Reusability

You can find useful notes in the SkillScanner repository.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2024/.