Hey Alexa, is this Skill Safe?: Taking a Closer Look at the Alexa Skill Ecosystem

Christopher Lentzsch*, Sheel Jayesh Shah†, Benjamin Andow‡§, Martin Degeling*, Anupam Das† and William Enck†
* Ruhr-Universität Bochum; {cl-imtm, martin.degeling}@ruhr-uni-bochum.de
† North Carolina State University; {sshah28, anupam.das, whenck}@ncsu.edu
‡ Google Inc.; andow@google.com

***Abstract*—Amazon's voice-based assistant, Alexa, enables users to directly interact with various web services through natural language dialogues. It provides developers with the option to create third-party applications (known as *Skills*) to run on top of Alexa. While such applications ease users' interaction with smart devices and bolster a number of additional services, they also raise security and privacy concerns due to the personal setting they operate in. This paper aims to perform a systematic analysis of the Alexa skill ecosystem. We perform the first large-scale analysis of Alexa skills, obtained from seven different skill stores totaling to 90,194 unique skills. Our analysis reveals several limitations that exist in the current skill vetting process. We show that not only can a malicious user publish a skill under any arbitrary developer/company name, but she can also make backend code changes after approval to coax users into revealing unwanted information. We, next, formalize the different skill-squatting techniques and evaluate the efficacy of such techniques. We find that while certain approaches are more favorable than others, there is no substantial abuse of skill squatting in the real world. Lastly, we study the prevalence of privacy policies across different categories of skill, and more importantly the policy content of skills that use the Alexa permission model to access sensitive user data. We find that around 23.3 % of such skills do not fully disclose the data types associated with the permissions requested. We conclude by providing some suggestions for strengthening the overall ecosystem, and thereby enhance transparency for end-users.**

I. Introduction

Voice-based computer interaction thrives on the ability to enable users to interact with devices and services through voice instead of keystrokes, mouse-movement or swipes. While speech recognition has been an active field of research for many years, it has seen widespread adoption recently. As a result there has been a rapid growth of voice-based web services such as Amazon Alexa [10]. Market research estimates that 3.25 billion devices with voice assistants are active today [32].

Amazon Alexa takes this opportunity to provide voice-based service as a platform and is the market leader in this area [30]. Developers can deploy applications that interact and provide functionality to end-users through Alexa enabled devices such as the Amazon Echo [11]. Such voice-based applications are called *skills* and are essentially apps that run on top of Amazon Alexa. Given that Amazon Echos are marketed for use at home and their microphones are continuously on, using voice-based third-party applications raise privacy concerns. Research shows that participants feel uncomfortable knowing that information from their private home has been shared or disclosed to third parties [40], [16], [36]. Moreover, recent studies continue to show increasingly sophisticated attacks on automated speech recognition systems [46], [20], [21] and on Alexa skills [56]. When Alexa integrates with other smart home IoT devices such as smart locks or smart cars,[1] security implications arise. An attacker can potentially expand her attack vector by deceiving a user to simply invoke skills that sound very similar to authentic skills. For example, 'lincoln way' (real skill) and 'lincoln weigh' (fictitious malicious skill) sound identical, but can potentially trick Alexa into activating the wrong skill and thereby enable the attacker to unlock a user's car. With Alexa's current policy of automatically enabling skills that match an invocation phrase, an adversary can potentially increase her odds of launching successful attacks.

Given the widespread adoption of Alexa and the potential for malicious actors to misuse skills, the goal of this paper is to perform a systematic analysis of the Alexa skill ecosystem and identify potential loopholes that can be exploited by malicious actors. In particular, we seek to answer the following broad research questions: **RQ1:** ***What limitations exist in the current skill vetting process?*** For this we thoroughly analyze the various steps involved in registering a skill, and identify potential flaws in the overall system. **RQ2:** ***How effective are skill squatting attacks?*** To address this question, we not only scan the skill stores to identify skills with phonetically similar invocation names, but also propose a *semi-automated* approach to test which skills Alexa actually activate when presented with potentially squatted skills. **RQ3:** ***Is the requirement of providing a privacy policy link effective?*** Alexa mandates a privacy policy link for skills that request certain permission APIs. We study the prevalence of privacy policies in different skill stores and analyze whether privacy policy links actually serve their purpose of informing users of their data practices.

In this paper, we perform a large-scale analysis of skills

§ This work was completed when the author was at IBM Research.

[1] Example of a skill that interacts with cars: https://amazon.com/Alexa-Skills-Smart-Home/b?ie=UTF8&node=14284863011, and with locks: https://amazon.com/Alexa-Skills-Smart-Home/b?ie=UTF8&node=14284863011

Network and Distributed Systems Security (NDSS) Symposium 2021
21-24 February 2021, San Diego, CA, USA
ISBN 1-891562-66-5
https://dx.doi.org/10.14722/ndss.2021.23111
www.ndss-symposium.org

collected across *seven* different stores and thoroughly study the whole skill ecosystem. We observe that a malicious actor can easily obtain sensitive information that is typically protected through a permission model by explicitly requesting such information from end users through the voice interface. We also see that an attacker can make stealthy changes to the backend code to coax a user into revealing information that is never invoked during the certification process. Interestingly, we also see that an attacker can register skills using well-known developer names, something that can further help an adversary to launch phishing attacks. Next, we find some evidence of skill squatting attempts, but in most cases such attempts are intentional and not malicious in nature, where the developer squats her own skills to improve the chance of the skills getting activated by Alexa. Lastly, we see that only a small portion of the skills actually link a privacy policy, and this situation does not improve even for skills under the 'kids' and 'health' categories, which often draw more attention under existing regulations such as COPPA [1], CCPA [48] and GDPR [2]. In summary, we make the following contributions:

- We perform the first large-scale analysis of Alexa skills across *seven* skills stores (US, UK, AU, CA, DE, JP, FR). We make our data available to the research community for further analysis. (§IV)
- We thoroughly analyze Amazon's skill certification process, and identify several potential loopholes that can be exploited by a malicious actor to publish deceptive skills. We also suggest guidelines for tightening up such loopholes. (§V)
- We identify common techniques used to squat skills, including one technique previously not discussed. We also design a semi-automated approach to gauge the effectiveness of various skill squatting techniques. We find that while some approaches are more successful than others, there is no substantial malicious abuse in the wild, and at times we see a developer squat her own skills to improve coverage. (§VI)
- Lastly, we analyze the privacy policy content of skills. On average only 24.2% of all skills provide a privacy policy link and skills in the 'kids' category are one of the biggest offenders. When contrasting skill permissions with privacy policies we find that 23.3% of the policies do not properly address the requested data types associated with the corresponding permissions. (§VII)

The remainder of this paper proceeds as follows. Section II provides background on Alexa skills and Amazon's skill certification process. Section III describes related work. Section IV describes datasets. Section V investigates skill vetting process (**RQ1**). Section VI investigates skill squatting (**RQ2**). Section VII studies privacy policies (**RQ3**). Section VIII discusses our recommendations. We conclude in Section IX.

II. Background

A. Building an Alexa Skill

Amazon opened Alexa to third-party developers in June, 2015 [44] to create an ecosystem similar to apps on mobile devices. There are two types of Alexa skills: native skills, developed and maintained by Amazon; and custom skills created by third-party developers. Custom skills must meet certain

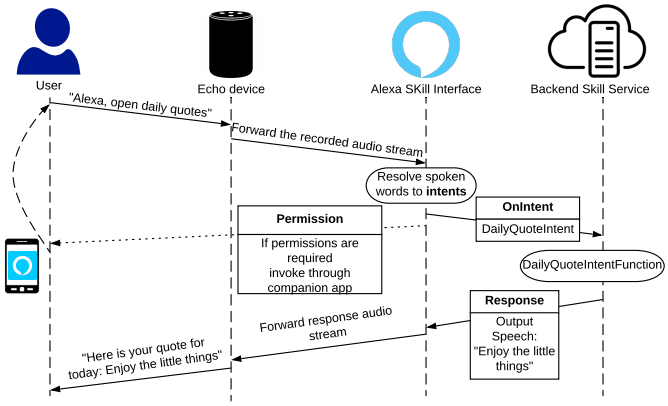


Fig. 1: Interactive workflow of an Amazon Alexa skill.

requirements and undergo an approval process. Figure 1 shows the overall data flow when using a skill. When a user speaks to an Alexa-enabled device, the audio is streamed to the Alexa web service. There, speech recognition and natural language processing techniques are used to identify phrases that match known skills published through the Alexa developer console.[2] Next, a structured JSON request is created and sent to a backend server registered with the matching skill (either hosted in AWS or on some external server). The server processes the request and responds accordingly. All speech recognition and conversion is handled by Alexa in the cloud [13], skills do not get access to raw audio data. Responses from skills are parsed by Alexa and are rendered using the same voice template for all skills. Every Alexa skill has an "interaction model" defining the words and phrases that users can utter to interact with the skill. This interaction model is analogous to a graphical user interface, where instead of clicking buttons and selecting options from dialog boxes, users make their requests and respond to questions by voice. The interaction model is defined when creating a custom skill. Following are elements required to build a custom skill [12]:

- An *invocation name* that identifies the skill. This name is used to initiate a conversation with the skill. Invocation names are not required to be globally unique. Alexa provides guidelines for selecting invocation names [7].
- A set of *intents* representing actions that users can invoke through the skill. An intent represents an action (triggering a backend handler) that fulfills a user's spoken request. For example, `AMAZON.HelpIntent` handles necessary actions when the user utters 'help'.
- A set of sample *utterances* that specify the words and phrases users can use to invoke the desired intents. These utterances are mapped to intents and this mapping forms the interaction model for the skill.
- A *cloud-based service* that accepts structured requests (i.e., intents in JSON format) and then acts upon them. This cloud-based service must be accessible over the Internet and defined as an endpoint when configuring the skill.
- A *configuration* that brings all of the above together, so Alexa can route requests to the desired skill. This configuration is created through the developer console [9].

Skill developers only have limited access to user data.

[2]See https://developer.amazon.com/alexa/console

As described above, requests are only forwarded to skills if they match the interaction model. Importantly, *utterances that enables a skill, but are followed by information that does not match any predefined intent, are not forwarded*. However, an adversary is capable of registering dormant intents to exfiltrate sensitive data, more details are provided in Section V-D. Users are also not directly identifiable as Amazon masks requests with identifiers that stay consistent for each skill, but across different skills the same user is assigned different identifiers.

B. Skill Certification Process

The Alexa developer console enables developers to test and submit their skills for verification before they are made public to end users. Once a skill is submitted for distribution, Amazon validates certain requirements. These certification requirements typically include [15]:

- Ensuring the skill meets the Alexa policy guidelines, which among many things includes making sure invocation names do not infringe existing brand names without providing proper affiliation.
- Performing all required voice interface and user experience tests, which include reviewing the intent schema and the set of sample utterances to ensure they are correct, complete, and adhere to voice design best practices.
- Performing all required functional tests, which includes checking whether the skill's basic functionality matches the information provided on the skill's description field.
- Ensuring the privacy policy link is a valid link. A privacy policy link is required if the skill requests access to sensitive data through the *permission model*.
- Ensuring the skill meets the security requirements for hosting services at external servers (i.e., non AWS Lambda servers), which includes checking whether the server responds to requests not signed by an Amazon-approved certificate authority.

Once a skill successfully passes all the validation steps, it officially appears in the skill store. Any changes made to the skill configuration and interaction model after the verification step will require the developer to re-initiate the whole verification process. However, modifications to backend code change does not trigger re-verification (this can be exploited by an attacker as discussed more in Section V-C).

III. Related Work

Attacks on speech recognition systems. As voice-based smart assistants have become more popular, we have also seen new attacks emerge against automated speech recognition systems (ASR). Several researchers have been successful in developing adversarial examples to trick voice-based interfaces. Carlini et al. [20] demonstrated how input audio can be synthesized in a way that it is unintelligible to humans, but are interpreted as commands by devices. In a followup study Carlini et al. [21] formalized a technique for constructing adversarial audio against Mozilla DeepSpeech with 100 % success rate. Vaidya et al. [51] were similarly successful in changing the input signal to fit a target transcription. More recently, Yuan et al. [53] showed that such hidden voice commands can be easily embedded into songs without being noticed by a human listener. Psychoacoustic models have also been used to manipulate acoustic signals such that it becomes imperceptible to humans [46]. Abdullah et al. [5] were able to exploit knowledge of the signal processing algorithms commonly used by voice processing systems (VPSecs) to successfully generate hidden voice commands. Furthermore, a series of independent studies have shown that it is possible to launch inaudible voice attacks by modulating hidden commands on ultrasound carriers [54], [47], [43]. However, attacks are mostly limited to lab settings and rarely work over the air, instead attacks are evaluated by directly feeding audio samples into the ASR models.

Attacks on skills. Edelman et al. [27] were the first to find thousands of domains with minor typographical variations on well-known web sites, a practice commonly known as "typosquatting". Their findings inspired a series of research towards measuring and mitigating the domain squatting threat [31], [42], [6], [50], [33]. Similarly, voice-squatting attacks have also been shown to be feasible with Alexa skills. Kumar et al. [35] first showed that skill squatting attacks can be launched when the invocation name of two different skills are pronounced similarly. Zhang et al. [55] recently introduced a new variant of the skill squatting attack where an attacker can use a paraphrased invocation name to hijack legitimate skills. This attack is based on the observation that Alexa favors the longest matching skill name when processing voice commands. In another concurrent work, Zhang et al. [56] design a linguistic-model-guided fuzzing tool to systematically discover the semantic inconsistencies in Alexa skills. They state that the developer controlled backend can be abused by the developer, for example by swapping legitimate audio files with malicious audio files. However, they do not provide details or demonstrate how this can be achieved.

Prevalence of privacy policy. In addition to the technical attack vectors to exfiltrate user data or execute commands on their behalf, there is also the possibility that skills themselves can try to trick users into exposing sensitive data. Legal regulations require companies to provide information to users about how they process personal data and for what purposes. Privacy policies have become the most important source for obtaining information about data practices. The importance of privacy policies for compliance with legal requirements has increased since the introduction of the European Union's General Data Protection Regulation (GDPR) [2]. A recent study by Degeling et al. [25] showed that the prevalence of privacy policies has increased to 85 % for websites, not limited to the European Union alone. However, several studies have shown the inconsistencies between what privacy policies state and what data is accessed [17]. For example, Libert [38] found that only 15 % of the information flowing from websites to third parties such as tracking and analytic services, is disclosed in the websites' privacy policies. Earlier, Zimmeck et al. [58] showed that 48 % of apps available in the Google Play store did not have a privacy policy even though the majority of the apps request access to at least one permission that would enable them to access personal data. In 2017, Alhadlaq et al. [8] performed a small analysis on Alexa skills (around 10,000 skills at the time) and found that 75 % of the skills did not have a privacy policy and 70 % of the existing policies did not mention anything specific to Alexa.

TABLE I: Comparison with existing work on Alexa skills. Symbols convey the following meanings – ○: not analyzed, ◐: partially analyzed, ●: analyzed.

	Zhang et al. [56]	Zhang et al. [55]	Kumar et al. [35]	Alhadlaq et al. [8]	Our work
Backend change	◐	○	○	○	●
Developer registration	○	○	○	○	●
Squatting	◐	◐	◐	○	●
Activation criteria	◐	◐	◐	○	●
Privacy policy	○	○	○	◐	●
Permission check	○	○	○	○	●

Distinction from prior work. In this paper, we present a *large-scale* systematic evaluation of the overall skill ecosystem, and identify flaws in the vetting process using proof-of-concepts. Table I highlights how our paper compares with other existing related works. We highlight ways in which the backend code can be updated to trigger dormant intents, which can deceive users into giving up sensitive data – something that has not been previously discussed or demonstrated. Zhang et al. [56] state that an attacker can swap backend audio files without providing concise details, whereas we demonstrate (by publishing a skill) how an attacker can register dormant intents of sensitive data types (Section V-C). We also showcase how an attacker can register skills using well-known developer names (e.g., Ring, Withings, Samsung) to deceive users into enabling phishing skills (Section V). We furthermore perform a large-scale empirical analysis to summarize the potential skill-squatting techniques/patterns observed in the wild; existing literature [55], [35] has mainly focused on showcasing how one specific approach can cause skill squatting. We also use a *semi-automated* approach to determine the efficacy of different squatting patterns — something that existing literature does not evaluate (Section VI). Lastly, we study skill privacy policies. Though, Alhadlaq et al. [8] provided an overview of privacy policy *availability*, our work is eight times larger than their analysis, covering categories that their overview missed (e.g., kids). Furthermore, while prior work stopped analysis at availability, we are the first to highlight potential COPPA violations, deficient enforcement of the privacy policy mandate, permission-to-policy inconsistencies, and root-cause analysis showing templates are causing potential violations of regulatory requirements (Section VII). We believe these findings are significant contributions over the prior work.

IV. Collecting Skill Data

A. Data Collection Setup

We collected data from Alexa skill stores [49] across seven countries: United States (US), United Kingdom (UK), Australia (AU), Canada (CA), Germany (DE), Japan (JP) and France (FR). We performed data collection in January, 2020. We used Selenium to automatically access a skill page and downloaded various information available on the page. To avoid geo-blocking, we crawled from servers located in data centers within each region of interest.[3] We first accessed all skills listed in different categories and also extracted additional skills listed under the "recommended skills" section of each page. The HTML files were parsed using Python to extract information about each skill's title, invocation name, required

[3]For example, the US skill Store is not accessible from an European-bound IP address.

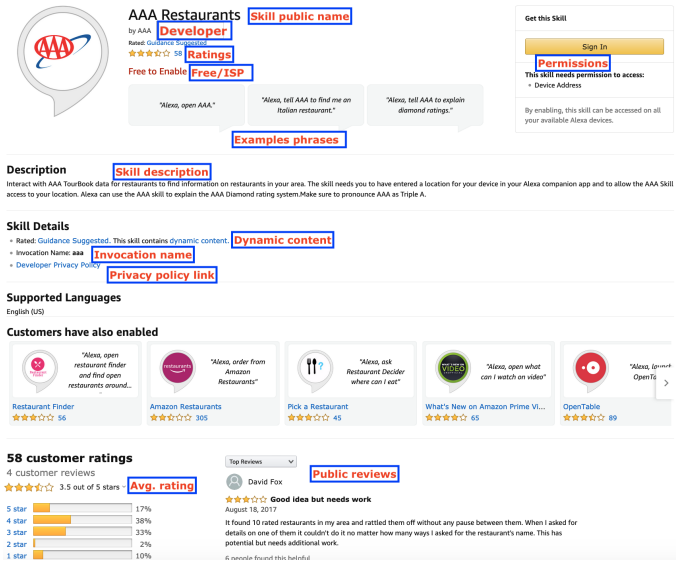


Fig. 2: Example of different information available on a skill's home page (unique skill identifier B071S69JDD).

TABLE II: Number of skills collected from different stores.

Country	# of Skills	# of Privacy Policies
US	58725	16733 (28.5 %)
UK	32218	6347 (19,7 %)
AU	21967	3946 (18.0 %)
CA	22298	4428 (19.9 %)
DE	10060	3098 (30.8 %)
JP	3336	1053 (31.6 %)
FR	2104	870 (41.3 %)
Combined	150,708	36,475 (24.2 %)

permissions, links to privacy policies, ratings and other details (an example of a skill's home page is shown in Figure 2). We honored Amazon's "robots.txt" restrictions and only downloaded the skill information pages. Still our attempts were limited by Amazon's API protection mechanism from time to time (less than 1 % of the requests), where we were redirected to a "Robot Check" website; we downloaded each such page after waiting for several minutes. The data is available to the research community for further analysis [4].

B. Brief Summary of Skill Metadata

All skill stores use the same taxonomy to organize skills into 21 different categories. Table II shows the number of skills available in each store. We collected a total of 150,708 skills listed across all stores, of which 90,194 were unique. The numbers exceed the 80,000 reported by Amazon [24] in 2019. Out of the unique skills 11,192 were available in all English-speaking skill stores (US, UK, CS, AU). Only 538 skills were common across the European skill stores (UK, FR, DE) and 163 skills were available in all seven countries. In terms of privacy policies we see that on average around 24.2 % of the skills provide a privacy policy link. Some of the European stores (like FR and DE) had relatively higher numbers of skills (30–40 %) with a privacy policy link.

We also looked at the overlap among the different stores. Table III summarizes the number of skills and developers overlapping across the different stores. We found 102 developers

TABLE III: Number of developers (in the upper half) and number of skills (in the bottom half) shared between different stores. The diagonal *bold numbers* represent number of skills and developers only available to the specific stores.

Store		# of common developers						
		AU	CA	DE	FR	JP	UK	US
# of common skills	AU	**(3023 / 948)**	7634	506	182	113	8164	8175
	CA	15151	**(2243 / 229)**	636	423	180	7838	8091
	DE	904	911	**(8558 / 2278)**	357	146	887	937
	FR	475	722	563	**(1189 / 499)**	120	440	455
	JP	234	246	226	196	**(3022 / 1019)**	191	247
	UK	16556	16815	1322	655	262	**(8557 / 2465)**	9796
	US	14916	16294	1295	601	299	19688	**(35698 / 13090)**

to publish skills to all the seven stores with "Invoked Apps LLC." [3] offering the highest number of skills (54 skills on average) in all stores. This number was significantly higher among English-speaking countries where a total of 5,567 developers publish in all four English-speaking stores. For the English-speaking stores, a developer by the name 'sachin nelwade' was the most prevalent publisher (over 400 skills in all of the English-speaking stores). These numbers suggest that while it is common for developers to publish skills in several stores, each store has its own set of unique developers and this is evident from the diagonal elements of Table III. Unless otherwise stated, we use the US dataset for most evaluations as it contains the most number of unique skills.

C. Research Ethics

To evaluate how Amazon's skill certification process works we created several skills for multiple purposes, e.g., registering skills under well-known company names and skills requesting phone numbers or zip codes from users verbally without registering a permission-protected intent, and testing activation of skills with identical invocation name. We created skills running on AWS Lambdas as well as ones backed by external endpoints. We did not collect any user data through the skills published, and we removed skills that could infringe a user's privacy immediately after they passed Amazon's certification process. We informed Amazon of our findings and they are currently conducting further investigation on them.

V. Loopholes in Skill Vetting

In this section, we answer **RQ1:** ***What limitations exist in the current skill vetting process?*** We perform a systematic analysis of the skill registration and certification process to identify potential pitfalls that might be exploitable by an attacker. First, we try to understand how Alexa selects a skill, among skills with the same invocation names. Next, we investigate if an attacker can register skills using any arbitrary developer/company name to potentially facilitate phishing attacks. We then test how easy it is for an adversary to trick users into revealing sensitive information by making backend code changes after a skill is certified. Lastly, we analyze how well Amazon mediates the collection of sensitive data.

A. Duplicate Skill Invocation Names

Over the years, Amazon has made it easier for users to enable Alexa skills. When Amazon first introduced Alexa, users had to enable skills either through the app or through their online account. In 2016, it became possible to explicitly enable skills with a voice command, and since mid 2017, Alexa

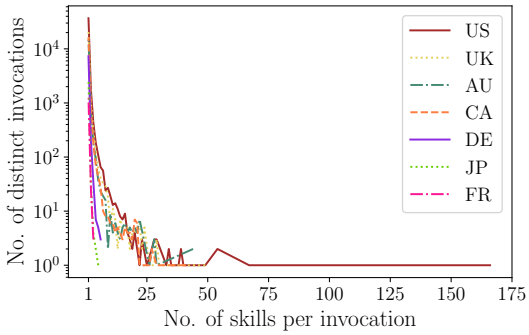


Fig. 3: Distribution of skills with the same invocation name across different stores. A large number of skills share the same invocation name.

now automatically enables skills if the user utters the right invocation name [39], [52], favoring native or first-party skills that are developed and maintained by Amazon [52]. Amazon, however, does not prevent non-native skills from sharing the same invocation name. Figure 3 shows the number of skills that have the same invocation phrases across the seven skill stores. From Figure 3, we see that a large number of skills share the same invocation name (as evident from the data points to the right of the line $x = 1$, which indicates that there are multiple skills with the same invocation name). In the US skill store, we found 9,948 skills that share the same invocation name with at least one other skill. Across all skill stores, we only found 36,055 skills with a unique invocation name. This makes it all the more important that when users install a skill by name, they get the skill they intend. Auto-enabling skills means that third-party developers can now target certain skills and register new skills with the *same* invocation phrase. For example, if you ask for "space facts" there are 81 such skills, of which Amazon automatically selects one. If the user's request does not match a skill's invocation name, Alexa automatically tries to fulfill the request by presenting the user with a list of probable skills to choose from [14]. Existing studies [56], [55], [35] have highlighted the existence of many duplicate skills, however, none of them have thoroughly analyzed how Alexa prioritizes among skills sharing the same invocation name.

The actual criteria that Amazon uses to auto-enable a skill among several skills with the same invocation names is unknown to the public. We, therefore, attempt to infer if certain skill attributes are statistically correlated with how Amazon prioritizes skills with the same invocation name. To understand the criteria Amazon uses to *auto-enable* a skill, we used a *semi-automated* approach to invoke skills with *duplicate* invocation names. To isolate the impact of the different attributes of a

TABLE IV: Fisher's exact test to determine attributes that are statistically correlated to skill activation.

Attribute 1	Attribute 2	# of skill pairs	Odds ratio	Favored attribute	p-value †
Different number of ratings		50	16	more ratings	< 0.0001 ****
Different avg. rating		50	5.44	higher avg. rating	0.00012 ***
Age of skill ‡		50	0.85		0.84162
Content advisory ◇		50	1.38		0.54874
Same number of ratings	Different avg. rating	29	3.61	higher avg. rating	0.03476 *
Same number of ratings	Age of skill	50	1.62		0.31734
Same number of ratings	Content advisory	50	1.17		0.84161

† *=$p < 0.05$, **=$p < 0.01$, ***=$p < 0.001$, ****=$p < 0.0001$; ‡ approximated; ◇ Content may include ads, nudity, religious intolerance or sexual themes.

skill, we only consider skill 'pairs', i.e., cases where only two skills (developed by two different developers) exist with the same invocation name, but has different other attributes. We analyzed the following attributes: 'number of ratings', 'average rating', 'age of skill' [4] and 'content advisory'.[5] We tested with 'number of ratings' and 'average rating' as developers have claimed *skill ratings* are used to auto-enable skills [23]. To determine if the publication date of a skill impacts the decision process we consider the 'age of skill' attribute and we also explore if the presence of 'content advisory' influences the decision of selecting one skill over another, assuming that Amazon may prefer a skill with more appropriate contents by default. Furthermore, we wanted to include 'number of permissions' attribute; however, we did not find sufficient samples (only 8 skill pairs) that differed in this attribute, thus a statistical analysis was not feasible.

We used Amazon's Text-to-Speech (TTS) service, 'Polly', to generate the samples and an Echo speaker (first generation) as receiver. We transmitted the invocation samples through a mono speaker in close distance and retrieved the list of skills activated from the Alexa app's activity panel. We repeated the experiment three times, each time with a newly created user profile with no interaction history. In our analysis a successful activation means the same skill was activated successfully across three different accounts. Next, for each attribute (or pair of attributes) that we test for statistical correlation, we randomly test skill pairs until we obtain 50 successful activations. Given that some skills were not functional at the time we ran our test, on average we ended up testing more than 50 skill pairs per attribute (or pair of attributes). Only in the case of testing skills with the same number of ratings, but different average rating were we able to test 29 skill pairs as there were no other skill pairs that fulfilled this requirement. Our analysis covered a total of 464 unique skills (232 unique invocation phrases) with successful activations across three user accounts.

We next conduct Fisher's exact test [28] for skills with different attributes to evaluate the impact of the respective attribute. Table IV highlights our findings. We found that skills with a higher number of ratings had an odds ratio of 16 with a $p - value < 0.0001$, i.e., skills with a higher number of ratings were 16 times more likely to be activated compared to the other skill with the same innovation name. We also found a higher average rating to be significant (odds ratio = 5.44 with $p - value < 0.001$). However, both the number of ratings and average rating are strongly correlated ($r = 0.65$, $p - value < 0.0001$), indicating skills with higher number of ratings tend to have higher average ratings. For the other two attributes: skill age and content advisory, we did not see any statistical significance. We then analyzed what other attributes excluding the most influential attribute (i.e., the number of ratings) might impact the prioritization process. We, therefore, only considered skill pairs with the same number of ratings, but different values for the other attributes. For cases where the number of ratings is the same, the skill with a higher average rating was more likely to be activated (odds ratio=3.61 and $p - value < 0.05$). Thus we see that the number of ratings and average rating are positively correlated with auto-enabling a skill. Note that we did not test all possible combinations of attributes as this would not scale in terms of obtaining sufficient samples to perform meaningful statistical tests.

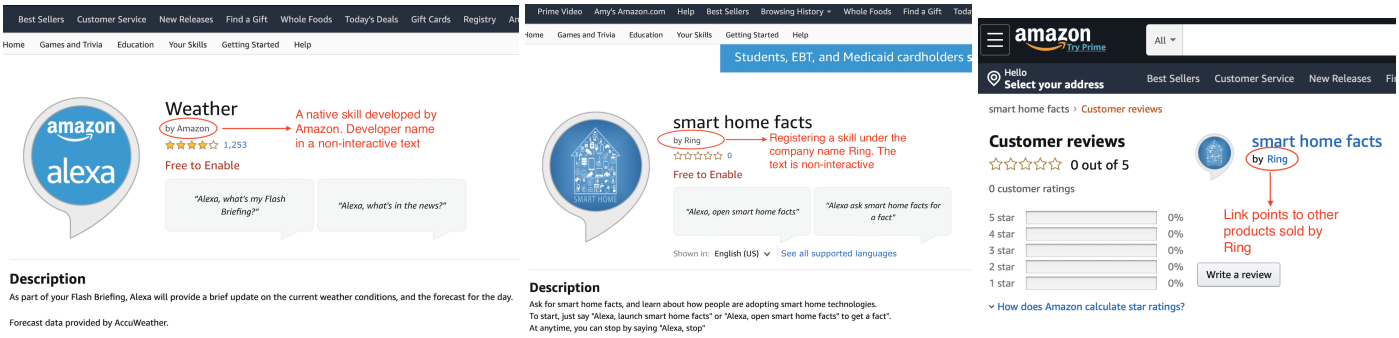
To investigate if the auto-enable feature can cause users to enable the wrong and at times risky skills, we created and published two fact skills with the same exact invocation name (B08FQY2KL8, B08G42WG9C). We made sure to register a unique invocation name not yet used by any skill in the US skill store. We first published one skill and tested whether the skill was activated across three accounts. Upon successful activation, we published the second skill (around 10 days later). In this skill (B08G42WG9C), users were first asked in which country they currently reside, so that the skill could provide more meaningful facts; thus emulating a skill accessing more sensitive data. We then reran the activation test with three new user accounts, where Alexa had two options to choose from. It turned out that the new skill (i.e., the one accessing more data) was automatically activated across the three accounts. This showcases how the auto-enable feature may lead to activating the wrong skill. Next, we attempted to see if providing reviews and ratings to the first skill (i.e., the one not accessing additional data) would influence the skill selection process. We recruited 12 volunteers to submit ratings (2-4 out of 5) and reviews for the skill which was not automatically enabled. The median difference in the number of ratings between skills (i.e., ones we tested) with the same invocation name was around 3, whereas the median number of ratings for a skill was 4. We then again reran the test after reviews and ratings were publicly available on Amazon.[6] However, we did not see Alexa switch between skills.

While our analysis on the public data shows correlation, it does not necessarily imply *causation*. For example, it is possible that the skill which is auto-enabled receives more

[4]We approximate the age of a skill using the metadata (last edit date) of the icon used by the skill. While this might not reflect the actual publish date, it can serve as an approximation as icons are not frequently changed.

[5]Content may include ads, nudity, religious intolerance or sexual themes.

[6]It took several days for all the ratings to be posted. We waited for two weeks before retesting.



(a) Native skill by Amazon **(b)** Our skill with 'Ring' as the developer **(c)** Amazon links to products by 'Ring'

Fig. 4: Screenshots for (a) a native skill by Amazon, (b) our own skill published as "Ring", and (c) developer name links to products manufactured by the same vendor. Attackers can register skills under different company names to facilitate phishing attacks through account linking.

reviews/ratings as the auto-enabled skill automatically appears on users' companion app and thereby makes it easy for users to provide ratings. This tells us that there are more deterministic factors other than ratings/reviews which Amazon internally uses and without knowing such factors it is infeasible for an attacker to manipulate the system.

Finding 1: *Due to the lack of transparency on how Amazon auto-enable skills with duplicate invocation names, users can easily activate the wrong skill*. While there is a positive correlation between a skill being activated and the number of ratings it receives, it does not imply causation as the auto-enabled skill appears on users' companion app and thereby making it easier for users to provide ratings.

B. Registering using Well-known Developer Names

When a skill is published in the skill store, it also displays the developer's name. We found that developers can register themselves with any company name when creating their developer's account with Amazon.[7] This makes it easy for an attacker to impersonate any well-known manufacturer or service provider. As Amazon displays the developer's name on a skill page, users can be easily deceived to think that the skill has been developed by an authentic source when it has really been published by an attacker. This can help an adversary launch phishing attacks especially for skills that require *account linking*.

To test to what extent Amazon validates developer information, we registered multiple skills using different well-known company names. For this purpose we registered fresh new Amazon developer accounts using well-known company names and submitted a skill for certification. We found that in most cases our skills were published without triggering any flags. For example, we were able to successfully register skills using "Microsoft", "Samsung", "Ring" and "Withings" as developer names. Figure 4 shows screenshots for one of our published skills. Interestingly, when viewing product reviews, Amazon updates the developer name (which is normally shown as a non-interactive text on the skill's information page) with a hyperlink for all products sold by the companies (shown on Figure 4c). This can further mislead users into believing that the skill was developed by an authentic company. However, our attempt in registering a skill with the developer name "Philips" was flagged as a potential infringement of the use of third-party trademark/brand. This tells us that there is *no consistent* approach to detect the registration of skills under different company names. Primarily, this is the outcome of manual vetting of skills by different employees, where one employee was able to detect our fraudulent registration attempt.

Finding 2: *An attacker can getaways with publishing skills using well-known company names*. This primarily happens because Amazon currently does not employ any automated approach to detect infringements for the use of third-party trademarks, and depends on manual vetting to catch such malevolent attempts which are prone to human error. As a result users might become exposed to phishing attacks launched by an attacker.

C. Code Change after Approval

Amazon sets requirements for hosting code in a back-end server that governs the logic of a skill. However, these requirements involve ensuring the backend server responds to only requests signed by Amazon. During the verification process, Amazon sends requests from multiple vantage points to check whether the server is responding to unsigned requests. However, no restriction is imposed on changing the backend code, which can change anytime after the certification process.

Currently, there is no check on whether the actual responses (logic) from the server has changed over time. Alexa, blindly converts the response into speech for the end-user. This can enable an attacker to stealthily change the response within the server without being detected. While this may sound benign at first, it can potentially be exploited by an adversary who intentionally changes the responses to trigger dormant, registered intents to collect sensitive data (e.g., phone number). Figure 5 highlights the overall flow diagram of how an attacker can exploit this gap to trick a user into giving up sensitive information. First, the attacker follows all the general steps (steps 1-3) for registering a skill, but inserts an intent(s) that will typically remain dormant under the benign case (i.e., the backend logic will not direct the user to trigger such intents).

[7]Providing developer name or company name is a one time process, and one cannot change the company name after it has been saved.

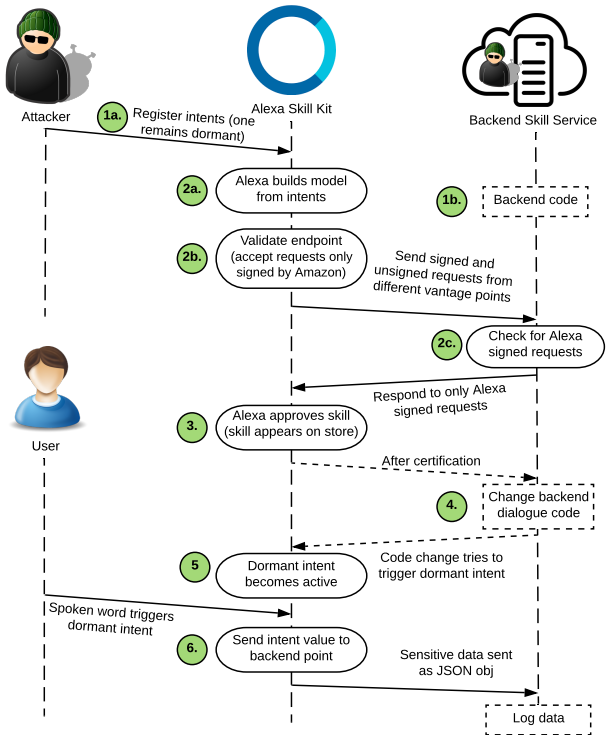


Fig. 5: Workflow diagram for making backend code change to trigger a dormant intent which will contain sensitive information like phone number.

Once the skill is published, the attacker then changes the backend logic to cause the user to invoke the dormant intent, which may correspond to some form of sensitive information such as phone number (steps 4-6).

We developed our own skill to test this approach where we built a trip planner skill asking a user to create a trip itinerary (B07N72MF9T). After the skill was published and tested, we changed the backend code, hosted as a Lambda service, to inquire the user for his/her phone number so that the skill could directly text (SMS) the trip itinerary. Note that during the initial certification process we did not ask users for their phone numbers, and hence when vetted by Amazon we reduced our chance of being flagged to request the phone number through their designated permission API. There are many other scenarios where this gap can be exploited. For example, a conversational skill targeted at kids can lure them into revealing details about their home or personal life after gaining their trust (assuming users' responses can trigger the skill's initial certified intents).

Finding 3: *An attacker can make code changes after approval to coax a user into revealing sensitive information.* This is possible as an attacker can register any number of intents during the certificate process, irrespective of whether or not all intents are triggered. Thus, an attacker can register dormant intents which are never triggered during the certification process to evade being flagged as suspicious. However, after the certification process the attacker can change the backend code (e.g., change the dialogue to request for a specific information) to trigger dormant intents.

TABLE V: Permission to data type mapping

Permission	Data Types
Device Address	postal address, city, country, zip code, state
Amazon Pay	person name, postal address, city, country, phone number, zip code, state
Postal Code	country, zip code, state
Location service	geographic location, speed, altitude, heading
Email Address	email address
First Name	person name
Full Name	person name
Mobile Number	phone number

D. Bypassing the Permission Model

Alexa skills can be configured to request permissions to access personal information, such as the user's address or contact information, from the Alexa account. Similar to permissions on smartphones, users enabling these skills must grant permission upon activation. These permissions can make interaction with a skill much more convenient, e.g., a weather skill with access to `device address` can report relevant weather forecasts based on the user's location. Permissions allow access to the following data types: `device address`, `customer name`, `customer email address`, `customer phone number`, `lists read/write`, `Amazon Pay`, `reminders`, `location services` and `skills personalization`. However, we found instances where skills bypass these permission APIs and directly request such information from end users. One could argue that this is not an issue as users explicitly provide their information, however, there may be a disconnect between how developers and users perceive the permission model. A user may not understand the difference between providing sensitive data through the permission APIs versus entering them verbally. Also, users may struggle to understand who is collecting the data as there is no distinction between the voice template used by native skills versus third-party skills (falsely assuming Amazon is protecting their data).

Skill developers can avoid requesting permissions to access personal data by simply requesting the personal data through verbal interactions. For example, we found several skills that included the name of specific locations as a part of the invocation phrase: in the German skill store, a forecasting service provides individual skills for 406 cities by appending the city names to the invocation phrase. In the US store, a local news provider named "Patch" has created 775 skills that include a city name. Such skills can potentially be used to track one's whereabouts.

A more concerning practice is when skill developers ask users for their personal data instead of requesting them through the permission API. Amazon relies on the *developer's declaration* of using the permission API instead of verifying a intent's *data type* itself. This way developers can bypass Amazon's requirement for providing a privacy policy when personal data is used (we study the efficacy of such privacy policies in Section VII-B). We tested this by building a skill that asks users for their phone numbers (one of the permission-protected data types) without invoking the `customer phone number` permission API. Even though we used the built-in data type of `Amazon.Phone` for the intent, the skill was not flagged for requesting any sensitive attribute. Unlike current loca-

TABLE VI: Detailed breakdown of skills potentially bypassing the Alexa permission model.

Filtering mechanism		Data Type				Unique skills *	w/o PP
		Name	Email	Phone	Location		
Skills detected through regular expression		432	417	242	416	1,482	668
After manually inspecting skill description		109	26	108	133	358	169
Activation	Verbally request data	65	4	33	76	166	99
	Non-verbally request data	1	1	1	0	3	2
	Does not request data	20	7	4	22	52	34
	Skill invocable but non-functional	19	12	62	24	113	25
	Skill not available in store	4	2	8	11	24	9

* Some skills access multiple data types, hence the summation across different data types will be slightly higher than the number of unique skills.

tion which may vary frequently over time, a phone number typically does not vary as frequently and hence should be instructed to be requested through the permission API.

To understand how prevalent this practice is in the skill store, we filtered skills that contain keywords related to different permissions (like 'address', 'location', 'postal', 'city', 'country', 'email', 'phone' , 'mobile', 'name', 'gps', 'state', 'zip') within their descriptions. A mapping of permissions to associated data types is shown in Table V. We found 13,499 such skills in the US store. We then performed a regular expression based search on the 13,499 skill *descriptions* to identify skills discussing the collection of privacy-sensitive data, protected by permissions. Note that our goal is to provide a *conservative lower-bound* approximation to demonstrate the existence of this practice rather than a comprehensive estimate of its prevalence. We segment each skill's description into sentence and leverage a set of four regular expressions conforming to the general pattern "*your <data_type>*" (shown in Table XI in Appendix A) to identify mentions of the user's name, phone number, location, and email address.

For each data type we removed skills that requested permission to access the corresponding data type. Table VI lists the number of candidate skills that initially matched our regular expressions. We then manually read the text to validate that the skills were actually discussing the use of such information and found many false positives due to reasons such as developers providing their email address and/or phone number as contact information, developers requesting access to sensitive data through account linking (this would require an additional authentication step), regional skills (often had 'city' and 'state' mentioned in the description) and skills requesting fixed player 'name' (e.g., gaming skills). After manually vetting the candidates we found a total of 358 unique skills potentially requesting information that is protected by a permission API. Next, to remove any remaining false positives, we manually *activated* the 358 skills to determine if they were really request data types protected by the permission APIs. Table VI shows the actual number of skills accessing data without using the dedicated permission API. We can see that the vast majority of the skills request data verbally (166 skills in total). However, a significant portion of the skills were also not functional, where either they were invocable but the backend server did not respond, or they were no longer available in the store. Table VII lists some of the non-verbal permission bypassing techniques. Interestingly, there are skills (B07QHB3P5L, B071F3BHBT) that request users to provide a name or email address through an external website (often associated with a passphrase or token to identify the user). We also found one skill requesting users to add phone numbers on a list created by the skill (B07HVHSP6W). Lastly, we looked at whether these skills were providing a privacy policy. From Table VI we see that around 59.8% (out of 169) of the active and functional skills bypassing the permission APIs (i.e., skills requesting data verbally or nonverbally) do not provide a privacy policy link. In terms of categories, we found that most of the skills bypassing the permission APIs belong to the 'Games & Trivia' category. Table XII in Appendix B lists the skill categories bypassing the permission APIs.

TABLE VII: Non-verbal permission bypassing techniques.

Bypassing Technique	Data
Redirects user to a website where they have to enter their name and game ID	Name
Redirects user to another website where they have to enter their email to generate a code which can then be used to create a game	Email
Skill asks user to add their phone number to a list created by the skill, which is then accessed by the skill	Phone

While these skills are likely benign (we can not definitively say if there was any malicious intent without knowing all possible ways in which the data is used), such capabilities can nevertheless be exploited by an attacker, especially when combined with Alexa's auto-enable feature to activate the wrong skill. Anecdotally, we found a skill providing insurance quotes that asks for other forms of personal data not protected by the permission APIs, such as DoB and gender. Worryingly, this skill does not provide a privacy policy.[8] In this paper, we focused on skills requesting data types protected by the permission model. Analyzing skills accessing all forms of sensitive data not protected by the permission model is something we leave as future work.

Finding 4: *Alexa does not properly mediate the intent of sensitive data types*. As demonstrated above an adversary can directly request data types that are structured to be protected by permission APIs. Even when the attacker uses a built-in data type, like `Amazon.Phone` for an intent, the skill does not get flagged for requesting sensitive data. This suggests that Amazon's permission model is somewhat flawed. While requesting different forms of sensitive information directly from the user rather than using a permission-protected API is not a technical implementation flaw, it is rather a conceptual flaw as users may struggle to understand who is collecting the data (there is no distinction between the voice template used by native skills versus third-party skills).

[8] https://www.amazon.com/dp/B07QJ5YFDH

VI. Skill Squatting

Given that we have shown the lack of transparency on how Alexa selects skills with same invocation phrases (in Section V-A), we next want to investigate **RQ2: *How effective are skill squatting attacks?*** While existing work (by Kumar et al. [35] and Zhang et al. [56]) has focused on demonstrating how one specific approach can squat skills, our investigation focuses on evaluating the *efficacy* of different squatting patterns found in the wild. We use a semi-automated approach where we use Amazon's TTS (Text-to-Speech) 'Polly' to generate utterances of invocation phrases that are phonologically very similar and record skills that get activated. This enables us to evaluate the efficacy of different squatting patterns — something existing literature[35], [56] has not analyzed.

A. Common Approaches for Squatting Skills

We use phonological distance between all pairs of unique invocation names (in the US store) to compute phonological similarity between invocation names. For this we first generated the phonetic encoding of each invocation name using the following three popular phonetic algorithms: *soundex* [41], *metaphone* [34] and *nysiis* [19]. We then computed the Levenshtein distance [37] between the phonetic encodes to determine similarity among invocation names. We also computed the generic Levenshtein distance among all invocation pairs. Figure 6 shows the CDF of the similarity among invocation names. We can see that most of the invocation names have similarity scores in the range of [0.2, 0.4]. However, for detecting potential voice-squatting skills we focused on the highly similar pairs. We, therefore, only considered invocation pairs with an average similarity score of ≥ 0.96 and marked them as *potential* squatting attempts.[9] We found 338 such invocation pairs. Next, we manually analyzed these invocation name pairs to filter pairs that sound quite different when pronounced (e.g., 'github stats' and 'github status'; 'indiana facts' and 'indian facts'). We eventually found 85 instances which we classified as *potential* squatting attempts. Note that we do not claim these skills as malicious squatting attempts; rather, they are ways in which Alexa may activate the wrong skill due to the auto-enable feature. Once we identified potential skill-squatting attempts, we shifted our focus on (manually) grouping them into different categories. Table VIII highlights the different patterns of squatting attempts found in the wild. The four common skill-squatting patterns are – homophones, punctuation, spacing, and different spellings (including spelling mistakes). Among these patterns, *homophones* and *different spellings* seem to be more prevalent. Interestingly, we also found *spacing* (i.e., joining or splitting words) as a technique, previously not discussed by existing literature.

To check for malicious intentions, we checked if developers systematically register skills to impersonate other skills. While we found few examples of skills providing similar functionality, we found no systematic large scale abuse. For example, in the US store the skill "i. p. lookup" (B01N13LZ7S) is homonym of "eye pee lookup" (B01GU5GE8A) — both skills provide the same functionality: a geo-lookup function for IPv4 addresses, but are registered with different developer name.

[9]We averaged all three phonetic encoding-based similarity scores to increase our odds of selecting truly similar invocation names. The threshold was empirically set to 0.96.

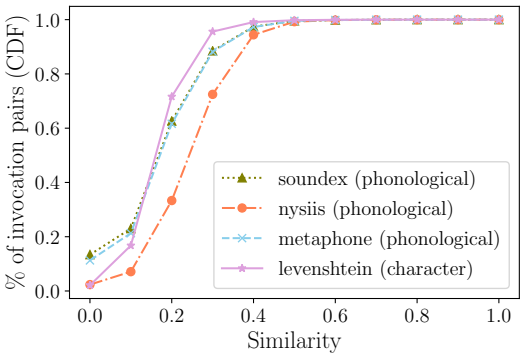


Fig. 6: Similarity score for invocation names taken from all English-speaking stores.

Similarly, the skills "mission two move" (B07HD8SSZG) and "mission too move" (B082WK7TNZ) are registered under different developer name. We also found several instances where the original skill developer registered multiple skills using similar invocation names. For example, the following two skills: "Sat Sri Akaal" (B07S18BCQ6) and "Sat Shri Akaal" (B07RY8RZDX) are registered by the same developer, likely to increase the probability of her skill being activated. However, across skill stores the registered homonyms were simply a variation between the British and American spelling (e.g., "colour lab" vs. "color lab").

Finding 5: *While we found four common approaches for squatting an existing skill, we did not find any systematic malicious abuse of skill squatting in the wild.* The non-evidence of malicious skill-squatting is a valuable data-point for the research community, as previous works have focused on showcasing how skills can be squatted without validating the prevalence and impact in the real world. However, it should be noted that the cause of non-detection could have been due to mitigation strategies enacted by Amazon, which may have been influenced by prior work.

B. Efficacy of Skill Squatting

To check to what extent the discovered squatting patterns work, we employed Amazon's TTS (Text-to-Speech) service named 'Polly' to generate utterances of invocation phrases using two user accounts. We use a similar setup as described in Section V-A, where we used an Amazon Echo as receiver and transmitted the samples with a mono speaker in close distance. We randomly selected skill pairs from the skill squatting patterns identified in the previous section (i.e, from Table VIII). We selected 10 such skill pairs using spelling variants, punctuation and homophones (30 pairs in total), plus the six pairs of word-spacing instances. We ran the TTS service for each skill pair, where one skill was invoked using the first account and the other using the second account. We logged Alexa's responses and observed the activity log on the app. Among the 36 skill pairs at least one of the skills was enabled across both two accounts in 28 cases. In eight cases, Alexa did not find a matching skill and tried to fulfill the request internally.

For the spelling variant scenario, in eight cases the same skill was enabled. Proper spelling seems to be preferred over mistakes (e.g., 'flick finder' over 'flic finder'), and American

TABLE VIII: Common Skill-squatting patterns based on the analysis of phonetically similar innovation names.

Description	Occurrences	Examples
Homophone: Similar or homophone utterances	32	"wierd facts" vs. "weird facts"; "hear motivation" vs. "here motivation"; "chuck norris fan" vs. "chack noris fan"
Punctuation: Invocations differed only in punctuation	18	"the rock of k. c." vs. "the rock of k c"; "cool one oh five" vs. "cool one o. five"; "farmer's market" vs. "farmers market"
Word-spacing: Compound words are joined or split differently	6	"world war two facts" vs. "worldwar two facts"; "under water sounds" vs. "underwater sounds"; "morning check list" vs. "morning check list"
Spelling: Different spellings or spelling mistake	29	"random colour" vs. "random color"; "travelling facts" vs. "traveling facts"; "recipe organizer" vs. "recipe organiser"

spelling over British spelling (e.g., 'recipe organizer' instead of 'recipe organiser'). For the remaining two pairs no matching skills were activated. Similar results were obtained for punctuation. In eight cases, the same skill was enabled (the remaining two were internally handled). In all of the succeeding test cases, invocation names without the use of punctuation was favored (e.g.,'farmers market' instead of 'farmer's market'). For the homophones, six skills were enabled across both accounts (the remaining four cases were internally handled) — favoring the original spelling of a word ('snake facts' instead of 'snek facts'). For the word-spacing variants, the joint words succeeded in five cases. Only for the case of 'world war two facts,' the variant with the additional space between 'worldwar' was preferred. These behaviors were consistent across both the accounts.

Finding 6: *Certain approaches within each skill-squatting pattern have a higher likelihood of successfully squatting skills*. For the different spelling types and homophones, we saw that correct/accepted spelling increased the likelihood of launching the expected skill over its variants with additional or altered letters. However, for punctuation appropriate usage reduced its chance of being activated. And for word-spacing, joint words succeeded most of the time.

VII. Privacy Policy Analysis of Skills

In this section, we answer **RQ3:** ***Is the requirement of a providing privacy policy link effective?*** Given that skills can register to collect a wide range of personal data, analyzing to what extent skills explicitly address such data in their privacy policies is an important issue. We are the first to study whether privacy policies of skills consistently disclose the data accessed and are compliant to existing regulations. We first highlight the prevalence of privacy policies in the different skill stores as not all skills are mandated to provide a privacy policy (Section VII-A). Next, we study the efficacy of the mandating privacy policies for skills requesting one or more permissions (Section VII-B).

A. Availability of Privacy Policies

Amazon enables skill developers to provide a privacy policy link addressing how data from end-users is collected and used. However, Amazon does not mandate a privacy policy for all skills, rather only for skills that request access to one or more of their permission APIs. We, therefore, first analyze the availability of privacy policy links in the US skill store. We found that around 28.5% of the US skills provide a privacy policy link (see Table IX), which is similar to what Alhadlaq et al. [8] reported back in 2017, when they found that around 25% skills out 11,827 skills provided a privacy policy link. We found that among all skills that provide a policy link around 2.9 % of them were not reachable in the US skill store. We even found a skill (B07DZT5YX9) with a policy link that pointed to "file://", referencing a document on the developers local machine. This indicates that Alexa, at times, is not properly vetting the privacy policy links.

TABLE IX: Number of skills per category in the US store along with the % of skills that have a privacy policy (PP).

Categories	# of skills	% of skills with PP
Smart Home	2,307	93.7 %
Connected Car	128	71.9 %
Social	1,372	37.2 %
News	5,629	43.3 %
Shopping	299	55.5 %
Productivity	1,050	39.2 %
Health & Fitness	1,980	42.2 %
Business & Finance	3,509	39.1 %
Music & Audio	6,762	38.1 %
Utilities	907	20.9 %
Sports	1,175	23.9 %
Food & Drink	1,377	29.6 %
Movies & TV	349	22.9 %
Local	166	19.3 %
Lifestyle	6,240	20.5 %
Weather	824	16.5 %
Travel & Transportation	1,178	16.9 %
Kids	1,887	13.6 %
Education & Reference	7,908	17.1 %
Novelty & Humor	3,361	12.0 %
Games & Trivia	10,201	14.9 %
Total	58,725	28.5 % (16,733)

The skill store allows us to browse available skills by categories which is same in all countries. Table IX lists the different categories and highlights the number of US skills in each category along with the percentage of skills that have privacy policies for each category. From Table IX, we see that a vast majority (93.7 %) of skills belonging to the 'smart home' category provide a privacy policy, followed by skills in the 'connected car' category. The categories that contain the least portion of skills with privacy policies include: 'game & trivia', 'novelty & humor' and 'education & reference' and 'kids'.

From a legal perspective, two categories are especially interesting: (1) the 'kids' category offering skills targeted towards children, and (2) the 'health and fitness' category that lists skills with medial facts or other health related services. Both COPPA [1] and EU's GDPR [2] require that consent be given by parents before kids interact with online services. Since Amazon is aware of this regulation, skill developers have to indicate if this skill is — "Directed to children under the age of 13 for the United States, as determined under the Children's Online Privacy Protection Act (COPPA)". Hints for developers

TABLE X: Number of skills requesting different permissions across the seven stores and the number of such skills without a privacy policy (shown in *bold*).

Permission	(# of skills / # of skills **w/o** privacy policy link)						
	US	UK	AU	CA	DE	JP	FR
Postal Code	**(492/3)**	(0/0)	(0/0)	(63/0)	**(77/2)**	(15/0)	(5/0)
Device Address	(446/0)	**(122/1)**	(47/0)	**(61/2)**	**(113/1)**	(19/0)	(13/0)
Lists Read	**(116/3)**	**(44/1)**	**(21/1)**	**(28/1)**	(28/0)	**(11/1)**	(8/0)
Lists Write	**(107/3)**	**(44/1)**	**(19/1)**	**(25/1)**	(31/0)	**(9/1)**	(7/0)
Notification *	(228/21)	(128/16)	(107/9)	(116/12)	(50/7)	(16/3)	(7/1)
Email Address	**(206/2)**	(76/0)	(46/0)	(51/0)	(53/0)	(23/0)	(21/0)
Full Name	(125/0)	(35/0)	(21/0)	(20/0)	(18/0)	(3/0)	(2/0)
Phone Number	(76/0)	(13/0)	(16/0)	(15/0)	(28/0)	(2/0)	(4/0)
Reminders *	(85/11)	(54/6)	(33/4)	(37/4)	(34/3)	(24/17)	(0/0)
First Name	**(50/3)**	(24/0)	(0/0)	(0/0)	(9/0)	(0/0)	(1/0)
Amazon Pay	(29/0)	(5/0)	(0/0)	(0/0)	(11/0)	(17/0)	(2/0)
Location Service	**(50/2)**	(24/0)	(10/0)	**(14/1)**	(10/0)	(2/0)	(5/0)
Any permission	(1464/41)	(428/24)	(235/14)	(311/19)	(324/13)	(120/21)	(55/1)

* these permissions do not mandate a privacy policy link

that help them decide whether or not their skills fall under this category are — "presence of child-oriented activities and incentives" and the "intended audience for the skill". If this box is checked, Alexa requires the skills to be enabled through the Alexa companion app, assuming the app is installed on the smartphone owned by the parent (who verifies herself as an adult by registering a credit card). Besides this one time consent, there are no further restrictions on kids' skills.

Finding 7: *In the US skill store only 13.6 % of skills belonging to the 'kids' category provide a privacy policy.* Interestingly, Amazon does not mandate a privacy policy for skills targeted towards children under the age of 13. The prevalence of privacy policies is somewhat higher for 'health and fitness' related skills (42.2 %). As privacy advocates we feel both 'kid' and 'health' related skills should be held to higher standards with respect to data privacy. The FTC is also closely observing skills in the 'kids' category for potential COPPA violations [26]. Research has provided evidence that guardians would also appreciate stricter controls [36].

B. Efficacy of Privacy Policy Requirement

Skills by default are not required to have any accompanying privacy policies. However, any skill requesting one or more permissions must have an accompanying privacy policy for it to be officially available in the skill store. While there are different legal constraints (e.g., GDPR, CCPA) in different geographic locations, the developer console does not have different requirements for developers in different countries (we verified this from both US and EU locations). Users enabling these skills must grant permission to these APIs upon activation. These permissions can make interaction with a skill much richer, e.g., a weather app with access to `device address` would know which location's weather to report when asked. The full list of permissions can be found in Table X, which shows the number of skills requesting different permissions. While the distribution of the permissions requested across various skill stores is different, we see that `device address` is prominently requested across all stores.

Figure 7 highlights the number of skills that request one or more permissions across various skill categories. We see that categories such as 'shopping', 'music and audio', 'business and finance', 'education and reference' and 'lifestyle' contain more skills that request access to different permissions. These categories of skills typically request access to `device address` and `postal code`. Interestingly, even though Amazon mandates developers to provide a privacy policy link when accessing these permission APIs (`notification` and `reminder` being the only exceptions), we found some instances (highlighted in *bold* in Table X) where privacy policy links were missing.

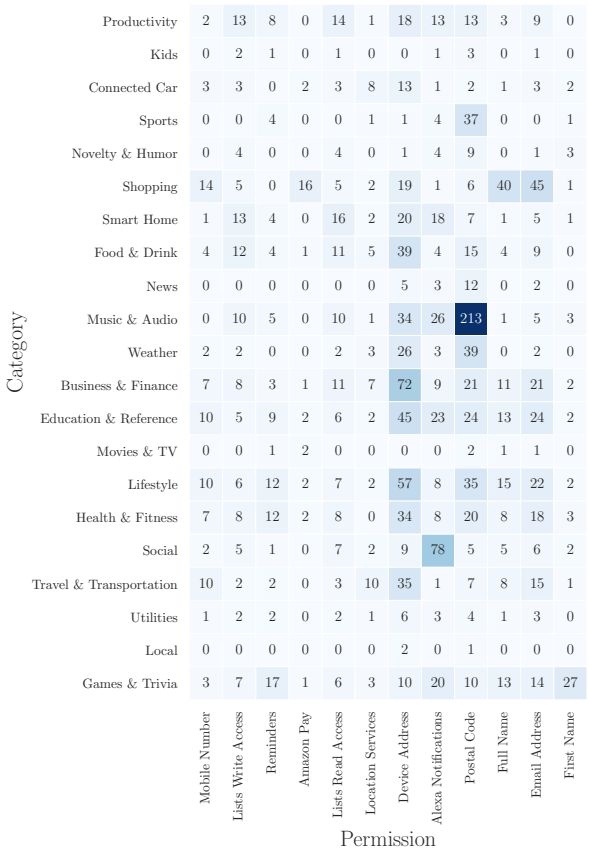


Fig. 7: Number of skills (US) that request specific permissions by store category.

Moreover, out of the 1,464 US skills requesting some form of permissions, 41 did not provide a policy link as they were requesting either the `notification` or `reminder` permission. For the remaining 1,423 skills we found that 1,285 skills (90%) provide a link posting content relevant to a privacy policy. We manually vetted all these privacy policies for this analysis. However, such process can be *automated* and to demonstrate that we designed a classifier to determine if the content of a privacy policy link was actually referring to a privacy policy. For this purpose, we manually vetted 1000 Android privacy policies [57] and 1000 non-privacy policy contents collected from blogging sites [45], Wikipedia and news articles [29]. We extracted TF-IDF of uni-grams and bi-grams from the text (first converting the text to lowercase and then removing all English stop words), and then used the TF-IDF features to train a SVM classifier (using 'sigmoid' kernel). Using 5-fold cross-validation we were able to obtain 99.8% precision and recall (accuracy was also around 99.8%). We then tested the privacy policies of skills requesting one

or more permissions as Amazon mandates these skills provide a privacy policy. The classifier had a precision and recall of 99.5% and 98.5%, respectively.

Finding 8: *90% of the US skills requesting one or more permissions actually provide a valid privacy policy*. The remaining 10% policy links mostly result in page not found, server errors or unregistered domains; however, some skills (30 such skills) point to homepages of websites, at times totally unrelated to the skill.

We also analyzed whether the privacy polices address the permissions requested. We contacted the authors of PoliCheck [18] and obtained the source code of their tool to measure flow-to-policy consistency analysis. As PoliCheck was designed to analyze data flows, we convert the eight permissions[10] that grant access to privacy-sensitive data into a set of first-party data flows based on a manually constructed mapping, as shown in Table V. For example, the `Postal Code` permission gives the skill access to the country, zip code, and state. Using the notation from PoliCheck,[11] we convert the `Postal Code` permission request into the following three data flows that denote first-party collection: (we, country), (we, zip code), and (we, state). Further, since we are applying PoliCheck to a new domain, we manually adapt their data type ontology (shown in Figure 9 in Appendix D) to include the data types covered by the Alexa permissions. We also performed trivial modifications to their code that identifies references to first-party entities within privacy policies by adding terms that refer to Alexa skills (e.g., "skills") and extending the synonym list for a set of entities.

PoliCheck classifies flow-to-policy consistency into two types of consistencies (i.e., clear, vague) and three types of inconsistencies (i.e., ambiguous, incorrect, omitted). In our case, we aim to measure whether the privacy policies disclose the permissions requested (permission-to-policy consistency). Therefore, after analysis with PoliCheck, we re-map the data flows and flow-to-policy consistency results back to the skill's permission requests. As this process may result in multiple consistency types being mapped back to the permission, we abstract PoliCheck's classification at a higher-level to either consistent or inconsistent for each data flow. When mapping back to permissions, we introduce the concept of *partial* consistency, which represents cases where the privacy policy only discloses a subset of the data types granted by a permission request. For example, consider a skill that requests the `Postal Code` permission, but only discloses they collect the user's country within the privacy policy. In this case, the `Postal Code` permission would be partially consistent with the policy, as it did not also disclose the collection of zip code and state. Pseudo code for the permission-to-policy consistency algorithm is provided in Appendix C.

Our initial dataset consists of 1,146 skills that request one or more of the eight permissions that grant access to privacy-sensitive data. We exclude 22 skills from the dataset whose privacy policy link does not directly display a privacy policy. We deploy the modified version of PoliCheck on these 1,124 skills with 1,447 permission requests, which produce 4,384 first-party data flows for analysis. We manually validate the consistency results from PoliCheck for the 4,384 data flows by using the validation methodology as documented in PoliCheck [18]. After manual validation, we found that PoliCheck correctly classified the data flows as either consistent or inconsistent with the privacy policy with an 83.3% precision. During validation, we noticed that most errors arose from the NER (Named-entity recognition) model not tagging entities (e.g., skill names) or missing sentence patterns for sentence classification, which can be addressed in future work with further domain adaption. Note that we corrected the misclassified flows during validation, so all results reported in the following findings are validated and correctly classified.

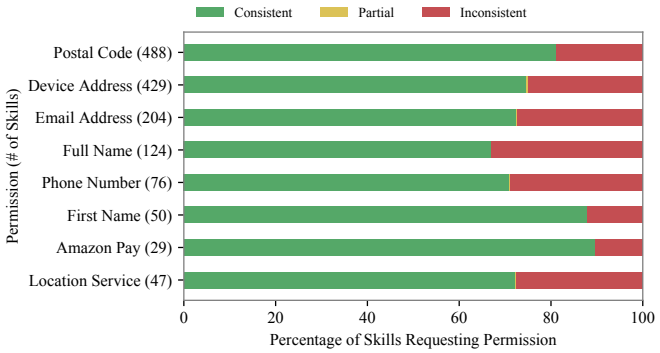


Fig. 8: Permission-to-policy consistency analysis results

Figure 8 shows the validated results of our permission-to-policy consistency analysis. In total, only 76.7% (862/1,124) of the privacy policies completely addressed all of their requested permissions. Note that 100 skills produced 404 errors when fetching the policy. We still included these in our analysis, as the lack of an available policy is equivalent to no disclosures at all. Surprisingly, 33.1% (41/124) of skills requesting the `Full Name` permission did not disclose the collection of such data in their privacy policy, which requires disclosure according to various regulations (e.g., CCPA [48], GDPR [2]). Several of these skills (B07MKPRVPB, B07RWWHK7W, B07MQKDKMZ, B07MFQH176) requesting the `Full Name` permission have privacy policies that explicitly state that they do not collect information from the user. For a set of 16 skills requesting the `Postal Code` and `Device Address` permissions (e.g., B072KL1S3G, B074PZQTXG, B07GKZ43J5), we found similarly potentially deceptive statements within the privacy policy ("We never collect or share personal data with our skills"). These cases may denote a misunderstanding by the developer on the purpose of providing a privacy policy and what they are required to disclose when accessing PII.

Two skills that requested the `Device Address` permission were marked as partially consistent (B076ZWH8ZL, B07VWR9YX8). However, their privacy policies only discuss requiring the state and country of the device, which may denote either that their privacy policies are incomplete or these skills are over-privileged and should request the more coarse-grained `Postal Code` permission.

[10]Discarding 'Reminder' and 'Notification' as they do not mandate a privacy policy. We also ignore 'List read/write' access as skills typically access lists created by themselves.

[11]PoliCheck[18] represents a data flow as (e, d), where data type d is flowing to entity e. First-party data flows are represented by setting e to "we."

Finding 9: *Around 23.3% of the privacy policies are not fully disclosing the data types associated with permissions requested by the skill*. Many skills (33.1%) accessing the `Full Name` permission did not disclose the collection of such data in their privacy policy.

We found that two widely-used privacy policy templates were resulting in 74 permission-to-policy inconsistencies across 46 skills. The BBB-Template was previously provided by the Better Business Bureau as a sample privacy policy template for websites.[12] We found 35 skills using the BBB-Template with 62 permission requests, such as `Device Address` (17 skills), `Email Address` (17 skills), and `Full Name` (15 skills). All 62 permission requests were marked as inconsistent. While the BBB-Template does discuss collection and sharing of data, it does not disclose the types or categories of data collected. For example, the BBB-Template includes overly broad statements, such as, "We are the sole owners of the information collected on this site. We only have access to/collect information that you voluntarily give us via email or other direct contact from you. We will not sell or rent this information to anyone." Privacy policies that solely discuss broad collection of "information" likely do not comply with the specificity requirement of disclosures defined by new regulations (e.g., CCPA [48], GDPR [2]).

The FPP-Template is a checkbox-based privacy policy generator provided by *freeprivacypolicy.com*. While the FPP-Template allows for a configurable specification of the data collection practices, we found that it was also a source of inconsistencies due to skills omitting data collection practices. This omittance of information can likely be attributed to developers not selecting all of the required checkboxes to cover their skill's behaviors or potential lack of expressibility by the generator. In total, we found 22 skills that used the FPP-Template requesting 31 permissions. In total, 12 permissions were marked as inconsistent across 11 skills that used the FPP-Template, such as `Device Address` (5 skills), `Postal Code` (5 skills), and `Phone Number` (1 skill).

Finding 10: *Privacy policy templates result in potential regulatory non-compliance in 46 skills*. The fact that developers are relying on these templates and they are resulting in permission-to-policy inconsistencies highlights an inherent flaw with the current publishing model of app markets. While developers are provided rich-APIs to develop their skills and obtain easy access to PII of end users, there does not appear to be any guidance to developers to create proper privacy policies. In turn, this negatively impacts the transparency of privacy risks placed on end users of these skills. While prior work [17] demonstrates that privacy policy templates are negatively impacting the transparency of privacy practices in the Android ecosystem, we demonstrate that this problem is also reflected in the Amazon Alexa skill ecosystem and is likely to be a problem in all application markets that similarly have a low barrier to entry.

[12]The sample template is no longer available on the Better Business Bureau's website (https://www.bbb.org/losangelessiliconvalley/for-businesses/understanding-privacy-policy/sample-privacy-policy-template/)

VIII. Discussion

Summary. We perform a comprehensive broad analysis of the Alexa skill ecosystem. This work advances the state-of-the-art by providing the following insights: (1) we highlight several gaps in the skill vetting process that can be exploited by an adversary; (2) we showcase that while common skill squatting techniques exist (we also found one new technique which we termed *word-spacing*) and are effective, there is no systematic abuse in the wild; (3) we show that 23.3% of the skills requesting permission to access sensitive user data do not fully disclose the data types associated with the permissions in their privacy policies. We open source our data to the research community to encourage further analysis in this domain [4].

A. Recommendations

Our analysis shows that while Amazon restricts access to user data for skills and has put forth a number of rules, there is still room for malicious actors to exploit or circumvent some of these rules. Auto-enabling skills reduces the distinction between native and third-party skills; however, users are still in the dark regarding which skill is responding to their queries. This can enable an attacker to exploit the trust they have built with the system. Based on our analyses we propose the following suggestions:

Skill-Type Indicator. Skill names and invocation phrases are not required to be unique. This design decision was made when skills required manual activation through the app, where users could see the description and developer name. Since Amazon introduced the auto-enable feature, users are less likely to know about the skills they are interacting with and how their data is being used. Alexa could, for example, provide some form of visual or verbal indicator (e.g., light or a different voice template) when interacting with a third-party application. Further HCI research is required to evaluate how voice assistants can ensure users are aware of what skills are being enabled.

Validating Developers. We have shown that it is possible to register accounts with any developer name, even those of well-known companies. This can mislead users and even be misused to launch phishing attacks. To improve the situation Amazon could utilize developer information to validate or flag trademark infringements. Also, like Google Play store Amazon can display developer details like contact email address or website for higher transparency.

(Recurring) Backend Validation. Currently, there is no provision to verify if the backend code has changed. A developer can push any code update once a skill has been approved without any further verification. While we do not expect Amazon to fully solve this problem as backend code may go through multiple rounds of updates, the threat needs to be acknowledged and understood. Potentially random recurring backend checks can be performed by Amazon.

Privacy Policy Template. Developers only need to provide a (working) policy link to get certified and start collecting user data. There is no check as to whether the policy link conveys all (or any) of the necessary information that a user might be interested in learning [22]. This issue can be addressed

by asking developers to fill out a simple policy template that will include what data is collected, for what purpose, for how long the data is retained, and whether users can delete or modify their data. Also, a valid contact address should be provided. Most of these requirements align with the minimum requirements imposed on companies/developers by GDPR and CCPA.

B. Limitations and Future Work

Our analysis has a few limitations. First, while our collection of skill data is the largest to the best of our knowledge, it is possible that we might have missed many skills. However, given that we have collected over 90,194 unique skills which exceeds the 80,000 reported by Amazon in 2019 [24], we do not foresee any significant difference in our reported numbers. Second, we provide a conservative lower-bound approximation to demonstrate the existence of skills bypassing the permission APIs, a more comprehensive estimate could be possible by utilizing more sophisticated NLP techniques. We plan to explore this in the near future. Lastly, in determining the effectiveness of different skill-squatting techniques, we tested a relatively small number of random skills. An *fully automated* approach would enable us to scale our test, significantly. However, developing such a fully automated approach is a challenging problem.

IX. Conclusion

In this paper, we analyze skills, which are third-party applications that are built on top of Alexa. While skills expand Alexa's capabilities and functionalities, it also creates new security and privacy risks. Our study covers skill stores from seven different countries with the goal to thoroughly analyze the overall vetting process enforced by Amazon. We identify several gaps in the current ecosystem that can be exploited by an adversary to launch further attacks, including registration of arbitrary developer name, bypassing of permission APIs, and making backend code changes after approval to trigger dormant intents. We also identify common skill squatting techniques, including one new technique. Moreover, we find that while certain skill-squatting techniques are more favorable, there is no systematic abuse of skill squatting in the wild. Lastly, we show that many skills requesting permissions do not properly address the use of such permission-protected data in their privacy policies. Based on our findings, we make several recommendations to strengthen the overall skill ecosystem.

Acknowledgement

We thank our anonymous reviewers for their feedback. This material is based upon work supported in parts by the National Science Foundation under grant number CNS-1849997 and by the state of North Rhine-Westphalia. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the state of North Rhine-Westphalia.

References

[1] "COPPA: Children's Online Privacy Protection Rule," 2019. [Online]. Available: http://uscode.house.gov/view.xhtml?req=granuleid%3AUSC-prelim-title15-section6501&edition=prelim

[2] "EU's General Data Protection Regulation," 2019. [Online]. Available: https://eugdpr.org/

[3] "Invoked apps," 2019. [Online]. Available: https://invokedapps.com/

[4] "A privacy & security analysis of the Alexa skill ecosystem," 2020. [Online]. Available: https://www.alexa-skill-analysis.org/

[5] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. B. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[6] P. Agten, W. Joosen, F. Piessensand, and N. Nikiforakis, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Proceedings of the 22nd Annual Network and Distributed System Security Symposium (NDSS)*, 2015.

[7] A. Alexa, "Choose the invocation name for a custom skill," 2019. [Online]. Available: https://developer.amazon.com/docs/custom-skills/choose-the-invocation-name-for-a-custom-skill.html

[8] A. Alhadlaq, J. Tang, M. Almaymoni, and A. Korolova, "Privacy in the Amazon Alexa Skills Ecosystem," in *10th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2017. [Online]. Available: https://petsymposium.org/2017/papers/hotpets/amazon-alexa-skills-ecosystem-privacy.pdf

[9] "Alexa developer console," Amazon, 2019. [Online]. Available: https://developer.amazon.com/alexa/console/ask

[10] "Alexa voice service," Amazon, 2019. [Online]. Available: https://developer.amazon.com/alexa-voice-service

[11] "All things Alexa," Amazon, 2019. [Online]. Available: https://www.amazon.com/Amazon-Echo-And-Alexa-Devices/b?ie=UTF8&node=9818047011

[12] "Understand custom skills," Amazon, 2019. [Online]. Available: https://developer.amazon.com/docs/custom-skills/understanding-custom-skills.html

[13] "Understand how users interact with skills," Amazon, 2019. [Online]. Available: https://developer.amazon.com/docs/ask-overviews/understanding-how-users-interact-with-skills.html

[14] "Understand name-free interactions," Amazon, 2019. [Online]. Available: https://developer.amazon.com/docs/custom-skills/understand-name-free-interaction-for-custom-skills.html

[15] "Certification requirements," Amazon Alexa, 2019. [Online]. Available: https://developer.amazon.com/en-US/docs/alexa/custom-skills/certification-requirements-for-custom-skills.html#submission-checklist

[16] T. Ammari, J. Kaye, J. Y. Tsai, and F. Bentley, "Music, Search, and IoT: How People (Really) Use Voice Assistants," *ACM Transactions on Computer-Human Interaction*, vol. 26, no. 3, pp. 1–28, Apr. 2019.

[17] B. Andow, S. Y. Mahmud, W. Wang, J. Whitaker, W. Enck, B. Reaves, K. Singh, and T. Xie, "PolicyLint: Investigating Internal Privacy Policy Contradictions on Google Play," in *Proceedings of the 28th USENIX Security Symposium (USENIX Security)*, 2019, pp. 585–602.

[18] B. Andow, S. Y. Mahmud, J. Whitaker, W. Enck, B. Reaves, K. Singh, and S. Egelman, "Actions Speak Louder than Words: Entity-Sensitive Privacy Policy and Data Flow Analysis with PoliCheck," in *Proceedings of the 29th USENIX Security Symposium (USENIX Security)*, 2020, pp. 985–1002.

[19] J. Burbige, "NYSIIS (New York State Identification and Intelligence System) parole and probation study-final report," 1970.

[20] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *Proceeding of the 25th USENIX Security Symposium (USENIX Security)*, 2016, pp. 513–530.

[21] N. Carlini and D. A. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," *CoRR*, vol. abs/1801.01944, 2018. [Online]. Available: http://arxiv.org/abs/1801.01944

[22] L. F. Cranor, "Necessary but not sufficient: Standardized mechanisms for privacy notice and choice," *J. on Telecomm. & High Tech. L.*, vol. 10, p. 273, 2012.

[23] P. Cutsinger, "How to improve Alexa skill discovery with name-free interaction and more," 2018. [Online]. Available: https://developer.amazon.com/blogs/alexa/post/0fecdb38-97c9-48ac-953b-23814a469cfc/skill-discovery

[24] M. Day, "Amazon's Alexa has 80,000 Apps—and No Runaway Hit," *Bloomberg Businessweek*, Mar. 2019. [Online]. Available: https://www.bloomberg.com/news/articles/2019-03-11/amazon-s-alexa-has-80-000-apps-and-no-runaway-hit

[25] M. Degeling, C. Utz, C. Lentzsch, H. Hosseini, F. Schaub, and T. Holz, "We Value Your Privacy ... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[26] Z. Doffman, "Amazon slammed for putting kids at risk with blatant violation of privacy laws," 2019. [Online]. Available: https://www.forbes.com/sites/zakdoffman/2019/05/09/amazons-echo-dot-kids-accused-of-violating-privacy-laws-and-putting-kids-at-risk/#4fb721a87e5a

[27] B. Edelman, "Large-scale registration of domains with typographical errors," 2003. [Online]. Available: https://cyber.harvard.edu/archived_content/people/edelman/typo-domains/

[28] R. A. Fisher, "On the interpretation of χ^2 from contingency tables, and the calculation of P," *Journal of the Royal Statistical Society*, vol. 85, no. 1, pp. 87–94, 1922.

[29] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, June 2018, pp. 708–719.

[30] Juniper Research, "Digital Voice Assistants in Use to Triple to 8 Billion by 2023, Driven by Smart Home Devices," Dec. 2018. [Online]. Available: https://www.juniperresearch.com/press/press-releases/digital-voice-assistants-in-use-to-triple

[31] M. T. Khan, X. Huo, Z. Li, and C. Kanich, "Every second counts: Quantifying the negative externalities of cybercrime via typosquatting," in *Proceedings of the 36th IEEE Symposium on Security and Privacy (SP)*, 2015, pp. 135–150.

[32] B. Kinsella, "Juniper estimates 3.25 billion voice assistants are in use today, Google has about 30% of them," 2019. [Online]. Available: https://voicebot.ai/2019/02/14/juniper-estimates-3-25-billion-voice-assistants-are-in-use-today-google-has-about-30-of-them/

[33] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis, "Hiding in plain sight: A longitudinal study of combosquatting abuse," in *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 569–586.

[34] M. Kuhn, "Metaphone searches," 1995. [Online]. Available: http://aspell.net/metaphone/metaphone-kuhn.txt

[35] D. Kumar, R. Paccagnella, P. Murley, E. Hennenfent, J. Mason, A. Bates, and M. Bailey, "Skill squatting attacks on Amazon Alexa," in *Proceedings of the 27th USENIX Security Symposium (USENIX Security)*, 2018, pp. 33–47.

[36] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, Are You Listening?: Privacy Perceptions, Concerns and Privacy-seeking Behaviors with Smart Speakers," *Proceedings. ACM Hum. Comput. Interact.*, vol. 2, no. CSCW, pp. 102:1–102:31, Nov. 2018.

[37] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[38] T. Libert, "An automated approach to auditing disclosure of third-party data collection in website privacy policies," in *Proceedings of the 27th World Wide Web Conference (WWW)*, 2018, pp. 207–216.

[39] T. Martin, "You can now use any Alexa skill without enabling it first," Mar. 2017. [Online]. Available: https://www.cnet.com/how-to/amazon-echo-you-can-now-use-any-alexa-skill-without-enabling-it-first/

[40] P. E. Naeini, S. Bhagavatula, H. Habib, M. Degeling, L. Bauer, L. Cranor, and N. Sadeh, "Privacy expectations and preferences in an IoT world," in *Proceedings of the 13th Symposium on Usable Privacy and Security (SOUPS)*. Usenix Association, 2017.

[41] "Soundex system: the soundex indexing system," National Archives and Records Administration, 2007. [Online]. Available: https://www.archives.gov/research/census/soundex.html

[42] N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, and W. Joosen, "Soundsquatting: Uncovering the use of homophones in domain squatting," in *Proceedings Information Security*, 2014, pp. 291–308.

[43] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2017, pp. 2–14.

[44] P. Sawers. (2015, Jun.) Amazon launches an SDK for developers to build on its Alexa voice-activated assistant. VentureBeat. [Online]. Available: https://venturebeat.com/2015/06/25/amazon-launches-an-sdk-for-developers-to-build-new-skills-for-amazon-echo/

[45] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker, "Effects of age and gender on blogging." in *AAAI spring symposium: Computational approaches to analyzing weblogs*, vol. 6, 2006, pp. 199–205.

[46] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[47] L. Song and P. Mittal, "Inaudible voice commands," *CoRR*, vol. abs/1708.07238, 2017. [Online]. Available: http://arxiv.org/abs/1708.07238

[48] "California Consumer Privacy Act (CCPA)," State of California Department of Justice, 2020. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[49] "Amazon Alexa: skill count in selected countries 2019," Statista, 2019. [Online]. Available: https://www.statista.com/statistics/917900/selected-countries-amazon-alexa-skill-count/

[50] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Felegyhazi, and C. Kanich, "The long "taile" of typosquatting domain names," in *Proceeding of the 23rd USENIX Security Symposium (USENIX Security)*, 2014, pp. 191–206.

[51] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, "Cocaine noodles: Exploiting the gap between human and machine speech recognition," in *Proceedings of the 9th USENIX Workshop on Offensive Technologies (WOOT)*, 2015.

[52] "Amazon is experimenting with Alexa skill auto-enablement," Voicebot, 2019. [Online]. Available: https://voicebot.ai/2018/03/22/amazon-experimenting-alexa-skill-auto-enablement/

[53] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *Proceedings of the 27th USENIX Security Symposium (USENIX Security)*, 2018, pp. 49–64.

[54] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 103–117.

[55] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian, "Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems," in *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 1381–1396.

[56] Y. Zhang, L. Xu, A. Mendoza, G. Yang, P. Chinprutthiwong, and G. Gu, "Life after speech recognition: Fuzzing semantic misinterpretation for voice assistant applications," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[57] S. Zimmeck, P. Story, D. Smullen, A. Ravichander, Z. Wang, J. Reidenberg, N. C. Russell, and N. Sadeh, "Maps: Scaling privacy compliance analysis to a million apps," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 3, pp. 66 – 86, 2019.

[58] S. Zimmeck, Z. Wang, L. Zou, R. Iyengar, B. Liu, F. Schaub, S. Wilson, N. Sadeh, S. M. Bellovin, and J. Reidenberg, "Automated Analysis of Privacy Requirements for Mobile Apps," in *Proceedings of the 24th Annual Network and Distributed System Security Symposium (NDSS)*, 2017.

APPENDIX A
REGULAR EXPRESSIONS FOR ANALYSIS OF SKILL DESCRIPTIONS

TABLE XI: Regular Expressions for Analysis of Skill Descriptions

Data Type	Regular Expression
Name	\b(your)\s+(((whole|entire|first/last|full|given|first|last|legal|first\sand\slast)\s+)? |(sur)?name)\b
Location	\b(your)\s+((home|work|personal|physical|billing|mailing|business|device('s)?)\s+)? (city|state|province| area|(postal\s+)?address|(zip|postal)\scode|((gps|device|geographic(al)?) \s+)?location| latitude|longitude|lat(itude)?/lon(gitude)?|lat(itude)?\sand\slon(gitude)? |region|country)\b
Phone Number	\b(your)\s+((home|work|personal|billing|business|device ('s)?)\s+)?(phone|telephone|mobile|cellular|cell (\s*phone)?)\s+number\b
Email	\b(your)\s+((home|work|personal|billing|business|valid| school|device('s)?)\s+)?((e|g)(\-)?mail(\saddress)?)\b

APPENDIX B
CATEGORIES OF SKILLS BYPASSING PERMISSION MODEL

TABLE XII: Category of skills potentially bypassing the Alexa permission API.

Bypassing technique	Category
Verbally request data	Games & Trivia (25), Lifestyle (19), Productivity (18), Education & Reference (17), Novelty & Humor (13), Travel & Transportation (11), Social (10), Weather (10), Health & Fitness (8), Food & Drink (8), Business & Finance (7), Kids (4), Movies & TV (3), Utilities (3), Music & Audio (2), News (2), Sports (2), Shopping (2), n (1), Smart Home (1)
Non-verbally request data	Games & Trivia (2), Social (1)
Does not request data	Lifestyle (10), Education & Reference (10), Novelty & Humor (8), Games & Trivia (8), Business & Finance (4), Health & Fitness (2), Weather (2), Social (2), Utilities (1), Local (1), Food & Drink (1), Smart Home (1), Movies & TV (1), Productivity (1)
Skill invocable but non-functional	Social (58), Business & Finance (10), Games & Trivia (9), Productivity (7), Smart Home (5), Education & Reference (4), Health & Fitness (3), Weather (3), Kids (2), Sports (2), Utilities (2), Shopping (2), Lifestyle (2), Novelty & Humor (1), Travel & Transportation (1), Local (1), Food & Drink (1)
Skill not available in store	Business & Finance (5), Travel & Transportation (4), Lifestyle (4), Education & Reference (2), Productivity (2), Kids (1), Games & Trivia (1), Novelty & Humor (1), Movies & TV (1), Health & Fitness (1), Social (1), Utilities (1)

APPENDIX C
SKILL PERMISSION-TO-POLICY CONSISTENCY ALGORITHM

Algorithm 1 Permission-To-Policy Consistency Algorithm

procedure PERMTOPOLICYCONSISTENCY($skill, privacyPolicy$)
  $results \leftarrow map[]$
  $flows \leftarrow []$
  while $p \leftarrow skill.permissions$ **do**
    while $t \leftarrow permToDataMap[p]$ **do**
      $flows.append(FirstPartyCollection(t))$
  while $pcheckRes \leftarrow PoliCheck(flows, privacyPolicy)$ **do**
    $dataType \leftarrow pcheckResult.dataType$
    while $perm \leftarrow dataToPermMap[dataType]$ **do**
      if $!hasPermission(skill, perm)$
        continue
      $results[perm].stmts.append(pcheckResult.stmts)$
      if $results[perm].consistency ==$ "PARTIAL"
        continue
      if $pcheckRes.consistency \in$ ["$clear$", "$vague$"]
        if $pcheckRes[perm].consistency ==$ "INCONS"
          $results[perm].consistency \leftarrow$ "PARTIAL"
          continue
        $results[perm].consistency \leftarrow$ "CONS"
      else
        if $pcheckRes[perm].consistency ==$ "CONS"
          $results[perm].consistency \leftarrow$ "PARTIAL"
          continue
        $results[perm].consistency \leftarrow$ "INCONS"
  return $results$

APPENDIX D
MODIFIED DATA TYPE ONTOLOGY

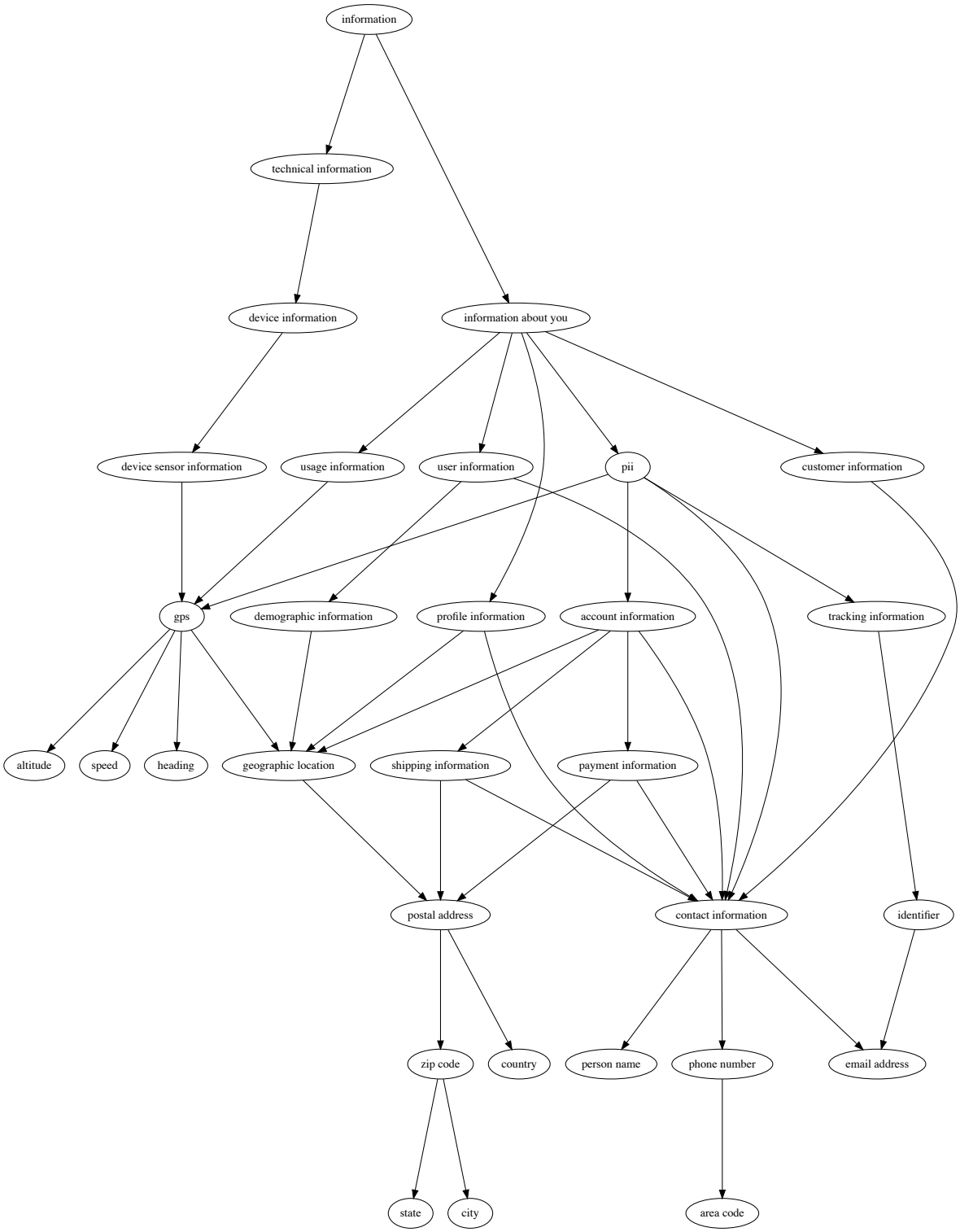


Fig. 9: Data type ontology modified for the Alexa domain